Stratix V Device Handbook

Volume 1: Device Interfaces and Integration

101 Innovation Drive
San Jose, CA 95134
www.altera.com

SV5V1-1.7
12.0

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

Contents

Chapter Revision Dates xi

Section I. Device Core

Chapter 1. Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices

Logic Array Blocks 1–1
LAB Interconnects 1–3
LAB Control Signals 1–3
Adaptive Logic Modules 1–4
ALM Operating Modes 1–7
Normal Mode 1–7
Extended LUT Mode 1–9
Arithmetic Mode 1–10
Shared Arithmetic Mode 1–11
ALM Interconnects 1–13
Clear and Preset Logic Control 1–13
LAB Power Management Techniques 1–14
Document Revision History 1–14

Chapter 2. Memory Blocks in Stratix V Devices

Overview 2–1
Embedded Memory Block Types 2–3
Parity Bit Support 2–3
Byte Enable Support 2–4
Packed Mode Support 2–5
Address Clock Enable Support 2–5
Mixed Width Support 2–7
Asynchronous Clear 2–7
Error Correction Code Support 2–8
Memory Modes 2–10
Single-Port RAM 2–10
Simple Dual-Port Mode 2–12
True Dual-Port Mode 2–14
Shift-Register Mode 2–15
ROM Mode 2–16
FIFO Mode 2–16
Clocking Modes 2–17
Independent Clock Mode 2–17
Input/Output Clock Mode 2–17
Read/Write Clock Mode 2–17
Single Clock Mode 2–18
Design Considerations 2–18
Selecting Embedded Memory Blocks 2–18
Conflict Resolution 2–18
Read-During-Write Behavior 2–19
Same-Port Read-During-Write Mode 2–19
Mixed-Port Read-During-Write Mode 2–20
Power-Up Conditions and Memory Initialization 2–21

Power Management 2–21
Document Revision History 2–22

Chapter 3. Variable Precision DSP Blocks in Stratix V Devices

Variable Precision DSP Block Overview 3–1
Operational Modes Overview 3–3
Variable Precision DSP Block Resource Descriptions 3–4
Input Registers Bank 3–5
Pre-Adder and Coefficient Select 3–7
Multipliers 3–8
Chainout Adder and Accumulator 3–8
Systolic Register 3–8
Output Register Bank 3–9
Operational Mode Descriptions 3–9
Independent Multiplier Modes 3–9
9 x 9, 16 x 16, 18 x18, 27 x 27, and 36 x 18 Multipliers 3–9
36-Bit Multiplier 3–13
Independent Complex Multiplier Modes 3–14
18 x 18 Complex Multiplier 3–14
18 x 25 Complex Multiplier 3–15
27 x 27 Complex Multiplier 3–16
Multiplier Adder Sum Mode 3–17
Sum of Square Mode 3–20
18 x 18 Multiplication Summed with 36-Bit Input Mode 3–20
Systolic FIR Mode 3–21
Variable Precision DSP Block Control Signals 3–22
Software Support 3–23
Document Revision History 3–24

Chapter 4. Clock Networks and PLLs in Stratix V Devices

Clock Networks in Stratix V Devices 4–1
Global Clock Networks 4–2
Regional Clock Networks 4–3
Periphery Clock Networks 4–3
Clock Sources Per Quadrant 4–7
Clock Regions 4–8
Entire Device Clock Region 4–8
Regional Clock Region 4–8
Dual-Regional Clock Region 4–8
Clock Network Sources 4–9
Dedicated Clock Input Pins 4–9
Internal Logic 4–9
DPA Outputs 4–9
HSSI Outputs 4–9
PLL Clock Outputs 4–9
Clock Input Pin Connections to GCLK and RCLK Networks 4–10
Clock Output Connections 4–12
Clock Control Block 4–12
Clock Enable Signals 4–15
Stratix V PLLs 4–16
PLL Migration Guidelines 4–24
Fractional PLL Architecture 4–25
Fractional PLL Usage 4–25

PLL External Clock I/O Pins . . . 4–25
PLL Control Signals . . . 4–27
pfdena . . . 4–27
areset . . . 4–27
locked . . . 4–27
Clock Feedback Modes . . . 4–28
Source Synchronous Mode . . . 4–28
Source Synchronous Mode for LVDS Compensation . . . 4–29
Direct Compensation Mode . . . 4–30
Normal Mode . . . 4–30
Zero-Delay Buffer Mode . . . 4–31
External Feedback Mode . . . 4–32
Implementing Multiple PLLs in Normal Mode and Source Synchronous Mode . . . 4–34
Clock Multiplication and Division . . . 4–36
Programmable Duty Cycle . . . 4–37
Clock Switchover . . . 4–37
Automatic Clock Switchover . . . 4–38
Manual Clock Switchover . . . 4–41
Guidelines . . . 4–41
PLL Reconfiguration and Dynamic Phase Shift . . . 4–42
Document Revision History . . . 4–43

Section II. I/O Interfaces

Chapter 5. I/O Features in Stratix V Devices

I/O Standard Support . . . 5–2
I/O Standards and Voltage Levels . . . 5–3
I/O Banks . . . 5–5
Modular I/O Banks . . . 5–6
I/O Structure . . . 5–8
3.3-V I/O Interface . . . 5–9
External Memory Interfaces . . . 5–9
High-Speed Differential I/O with DPA Support . . . 5–9
Current Strength . . . 5–10
Slew-Rate Control . . . 5–11
I/O Delay . . . 5–11
Programmable IOE Delay . . . 5–11
Programmable Output Buffer Delay . . . 5–11
Open-Drain Output . . . 5–12
Bus-Hold . . . 5–12
Pull-Up Resistor . . . 5–12
Pre-Emphasis . . . 5–12
Differential Output Voltage . . . 5–13
MultiVolt I/O Interface . . . 5–13
OCT Support and I/O Termination Schemes . . . 5–14
R_S OCT Without Calibration . . . 5–14
R_S OCT with Calibration . . . 5–15
R_T OCT with Calibration . . . 5–17
Dynamic OCT . . . 5–17
LVDS Input R_D OCT . . . 5–18
Summary of OCT Assignments . . . 5–19
OCT Calibration . . . 5–20
Sharing an OCT Calibration Block on Multiple I/O Banks . . . 5–20

OCT Calibration Block Modes of Operation 5–22
Power-Up Mode 5–22
User Mode 5–22
OCT Calibration 5–23
Serial Data Transfer 5–24
Example of Using Multiple OCT Calibration Blocks 5–25
Termination Schemes for I/O Standards 5–25
Single-Ended I/O Standards Termination 5–25
Differential I/O Standards Termination 5–28
LVDS 5–30
Differential LVPECL 5–31
RSDS 5–31
Mini-LVDS 5–33
LVDS Direct Loopback Mode 5–34
Design Considerations 5–34
I/O Bank Restrictions 5–34
Non-Voltage-Referenced Standards 5–35
Voltage-Referenced Standards 5–35
Mixing Voltage-Referenced and Non-Voltage-Referenced Standards 5–35
V_{CCPD} Restriction 5–35
LVDS Channels 5–36
Document Revision History 5–36

Chapter 6. High-Speed Differential I/O Interfaces and DPA in Stratix V Devices

Overview 6–1
Locations of the I/O Banks 6–3
LVDS Channels 6–3
LVDS SERDES 6–6
Differential Transmitter 6–7
Programmable Differential Output Voltage (V_{OD}) and Programmable Pre-Emphasis 6–9
Programmable V_{OD} 6–10
Programmable Pre-Emphasis 6–11
Differential Receiver 6–12
Differential I/O Termination 6–13
Receiver Hardware Blocks 6–14
DPA Block 6–14
Synchronizer 6–15
Data Realignment Block (Bit Slip) 6–15
Deserializer 6–17
Receiver Datapath Modes 6–18
Non-DPA Mode 6–18
DPA Mode 6–20
Soft-CDR Mode 6–21
LVDS Direct Loopback Mode 6–22
LVDS Interface with the Use External PLL Option Enabled 6–23
PLLs and Stratix V Clocking 6–27
Source-Synchronous Timing Budget 6–28
Differential Data Orientation 6–28
Differential I/O Bit Position 6–28
Transmitter Channel-to-Channel Skew 6–30
Receiver Skew Margin for Non-DPA Mode 6–30
Differential Pin Placement Guidelines 6–35
Guidelines for DPA-Enabled and DPA-Disabled Differential Channels 6–35
DPA-Enabled Channels, DPA-Disabled Channels, and Single-Ended I/Os 6–35

Document Revision History 6–36

Chapter 7. External Memory Interfaces in Stratix V Devices

Memory Interface Pin Support 7–3
Using the RZQ Pins in a DQ/DQS Group for Memory Interfaces 7–9
Stratix V External Memory Interface Features 7–9
DQS Phase-Shift Circuitry 7–10
Delay-Locked Loop 7–11
Phase Offset Control 7–17
PHY Clock (PHYCLK) Networks 7–18
DQS Logic Block 7–20
DQS Delay Chain 7–21
Update Enable Circuitry 7–21
DQS Postamble Circuitry 7–22
Leveling Circuitry 7–23
Dynamic On-Chip Termination Control 7–24
I/O Element Registers 7–25
Delay Chain 7–27
I/O Configuration Block and DQS Configuration Block 7–29
Document Revision History 7–32

Section III. System Integration

Chapter 8. Hot Socketing and Power-On Reset in Stratix V Devices

Stratix V Hot-Socketing Specifications 8–1
Stratix V Devices Can Be Driven Before Power Up 8–2
I/O Pins Remain Tri-Stated During Power Up 8–2
Insertion or Removal of a Stratix V Device from a Powered-Up System 8–2
Hot-Socketing Feature Implementation in Stratix V Devices 8–3
Power Sequencing Requirements 8–4
Power-On Reset Circuitry 8–5
Power-On Reset Specifications 8–7
Document Revision History 8–8

Chapter 9. Configuration, Design Security, and Remote System Upgrades in Stratix V Devices

Configuration Features 9–2
Power-On Reset Circuit and Configuration Pins Power Supply 9–2
POR Delay Specification 9–2
Power-On Reset Circuit 9–3
V_{CCPGM} Pin 9–3
V_{CCPD} Pin 9–3
Configuration Sequence 9–4
Power Up 9–4
Reset 9–4
Configuration 9–5
Configuration Error 9–5
Initialization 9–5
User Mode 9–6
Configuration Schemes 9–7
MSEL Pin Settings 9–7
Raw Binary File Size 9–8
Fast Passive Parallel Configuration 9–9
DCLK-to-DATA[] Ratio for FPP configuration 9–9

FPP Multi-Device Configuration . . . 9–11
FPP Configuration Timing . . . 9–13
Active Serial Configuration (Serial Configuration Devices) . . . 9–17
AS Multi-Device Configuration . . . 9–20
AS Connection Guidelines . . . 9–23
AS Configuration Timing . . . 9–23
Estimating the Active Serial Configuration Time . . . 9–24
Programming EPCS and EPCQ . . . 9–24
Passive Serial Configuration . . . 9–29
PS Configuration Using a MAX II Device, MAX V Device, or Microprocessor . . . 9–29
PS Configuration Timing . . . 9–32
PS Configuration Using a Download Cable . . . 9–33
Multi-Device PS Configuration Using Download Cable . . . 9–35
JTAG Configuration . . . 9–36
CONFIG_IO Instruction . . . 9–38
Multi-Device JTAG Configuration . . . 9–38
Device Configuration Pins . . . 9–40
Configuration Data Decompression . . . 9–44
Remote System Upgrades . . . 9–46
Configuration Image Types . . . 9–47
Remote Update Mode . . . 9–48
Remote System Upgrade Using EPCQ 256 . . . 9–49
Dedicated Remote System Upgrade Circuitry . . . 9–49
Remote System Upgrade Registers . . . 9–51
Control Register . . . 9–51
Status Register . . . 9–52
Remote System Upgrade State Machine . . . 9–53
User Watchdog Timer . . . 9–53
Enabling the Remote System Update Feature . . . 9–54
ALTREMOTE_UPDATE Megafunction . . . 9–55
Design Security . . . 9–55
JTAG Secure Mode . . . 9–57
Security Key Types . . . 9–57
Security Modes . . . 9–58
Design Security Implementation Steps . . . 9–59
Document Revision History . . . 9–60

Chapter 10. SEU Mitigation in Stratix V Devices

Error Detection Fundamentals . . . 10–1
Configuration Error Detection . . . 10–2
User Mode Error Detection and Correction . . . 10–2
Error Detection Pin Description . . . 10–5
Error Detection Block . . . 10–6
Error Detection Registers . . . 10–7
Error Detection Timing . . . 10–8
Software Support . . . 10–11
Recovering From CRC Errors . . . 10–11
Document Revision History . . . 10–12

Chapter 11. JTAG Boundary-Scan Testing in Stratix V Devices

IEEE Std. 1149.6 Boundary-Scan Register . . . 11–2
BST Operation Control . . . 11–3
I/O Voltage Support in a JTAG Chain . . . 11–5

Boundary-Scan Description Language Support 11–6
Document Revision History 11–6

Chapter 12. Power Management in Stratix V Devices

Stratix V Programmable Power Technology 12–2
Stratix V External Power Supply Requirements 12–3
Temperature Sensing Diode 12–3
Internal Temperature Sensing Diode 12–4
External Temperature Sensing Diode 12–4
Document Revision History 12–6

Additional Information

How to Contact Altera Info–1
Typographic Conventions Info–1

Chapter Revision Dates

The chapters in this document, Stratix V Device Handbook *Volume 1*, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51002-1.4*

Chapter 2. Memory Blocks in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51003-1.4*

Chapter 3. Variable Precision DSP Blocks in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51004-1.4*

Chapter 4. Clock Networks and PLLs in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51005-1.4*

Chapter 5. I/O Features in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51006-1.5*

Chapter 6. High-Speed Differential I/O Interfaces and DPA in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51007-1.4*

Chapter 7. External Memory Interfaces in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51008-1.4*

Chapter 8. Hot Socketing and Power-On Reset in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51009-1.4*

Chapter 9. Configuration, Design Security, and Remote System Upgrades in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51010-1.7*

Chapter 10. SEU Mitigation in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51011-1.5*

Chapter 11. JTAG Boundary-Scan Testing in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51012-1.5*

Chapter 12. Power Management in Stratix V Devices
Revised: *June 2012*
Part Number: *SV51013-1.3*

Section I. Device Core

This section describes the Stratix® V device family core, which is the most architecturally advanced, high-performance, low-power FPGA in the marketplace. This section includes the following chapters:

- Chapter 1, Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices
- Chapter 2, Memory Blocks in Stratix V Devices
- Chapter 3, Variable Precision DSP Blocks in Stratix V Devices
- Chapter 4, Clock Networks and PLLs in Stratix V Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

1. Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices

SV51002-1.4

This chapter describes the features of the logic array blocks (LABs) in the Stratix® V core fabric. LABs are made up of adaptive logic modules (ALMs) that you can configure to implement logic functions, arithmetic functions, and register functions. LABs and ALMs are the basic building blocks of the Stratix V device. ALMs provide advanced features with efficient logic utilization and are completely backward-compatible.

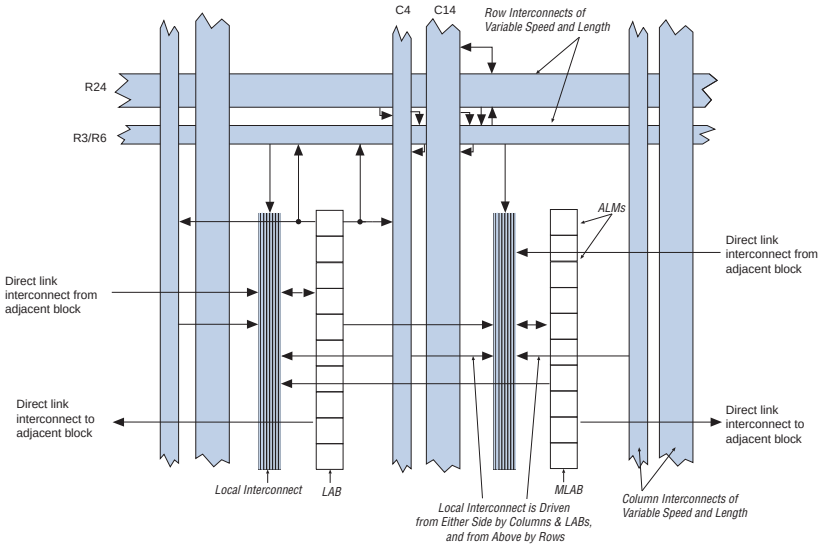
This chapter contains the following sections:

- "Logic Array Blocks" on page 1–1
- "Adaptive Logic Modules" on page 1–4

Logic Array Blocks

Each LAB consists of ten ALMs, various carry chains, shared arithmetic chains, control signals, and a local interconnect. The local interconnect transfers signals between ALMs in the same LAB. The direct link interconnect enables the LAB to drive into the local interconnect of its left and right neighbors. The Quartus® II Compiler places associated logic in the same LAB or adjacent LABs, allowing the use of local and shared arithmetic chain for performance and area efficiency. Figure 1–1 shows the Stratix V LAB structure and the LAB interconnects.

Figure 1–1. LAB Structure for Stratix V Devices


C4
C14
Row Interconnects of Variable Speed and Length
R24
R3/R6
ALMs
Direct link interconnect from adjacent block
Direct link interconnect from adjacent block
Direct link interconnect to adjacent block
Direct link interconnect to adjacent block
Local Interconnect
LAB
MLAB
Local Interconnect is Driven from Either Side by Columns & LABs, and from Above by Rows
Column Interconnects of Variable Speed and Length

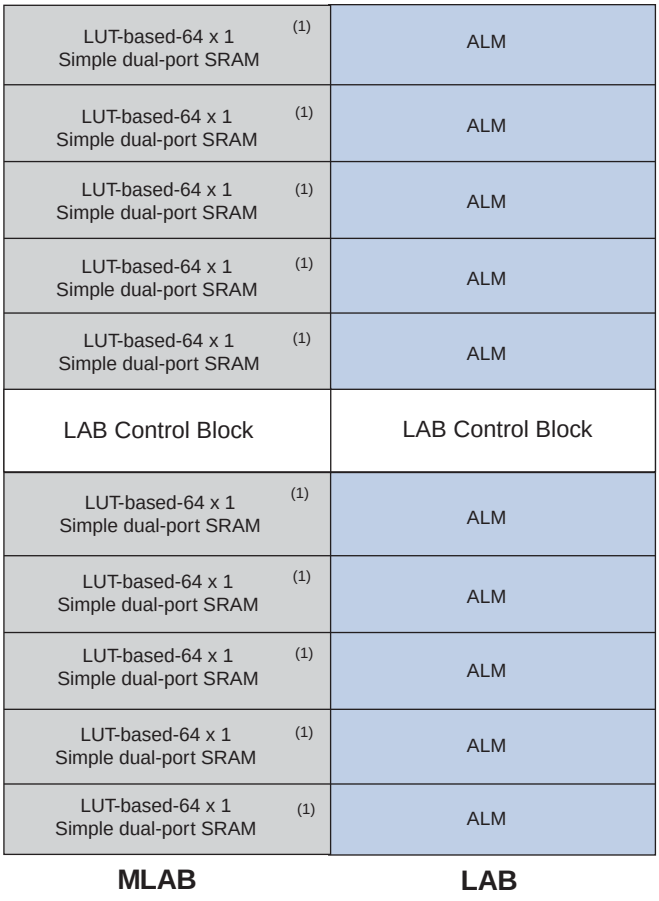
© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

The memory LAB (MLAB) is a derivative of the Stratix V LAB. The MLAB adds look-up table (LUT)-based SRAM capability to the LAB, as shown in Figure 1–2. The MLAB supports a maximum of 640 bits of simple dual-port SRAM. You can configure each ALM in an MLAB as either a 64 × 1 or a 32 × 2 block, resulting in a configuration of either a 64 × 10 or a 32 × 20 simple dual-port SRAM block. MLAB and LAB blocks alternate in Stratix V devices. Therefore, the maximum number of available MLABs is half of the total number of LABs. The MLAB is a superset of the LAB and includes all LAB features.

For more information about MLABs, refer to the *Memory Blocks in Stratix V Devices* chapter.

Figure 1–2. LAB and MLAB Structure for Stratix V Devices



Note to Figure 1–2:

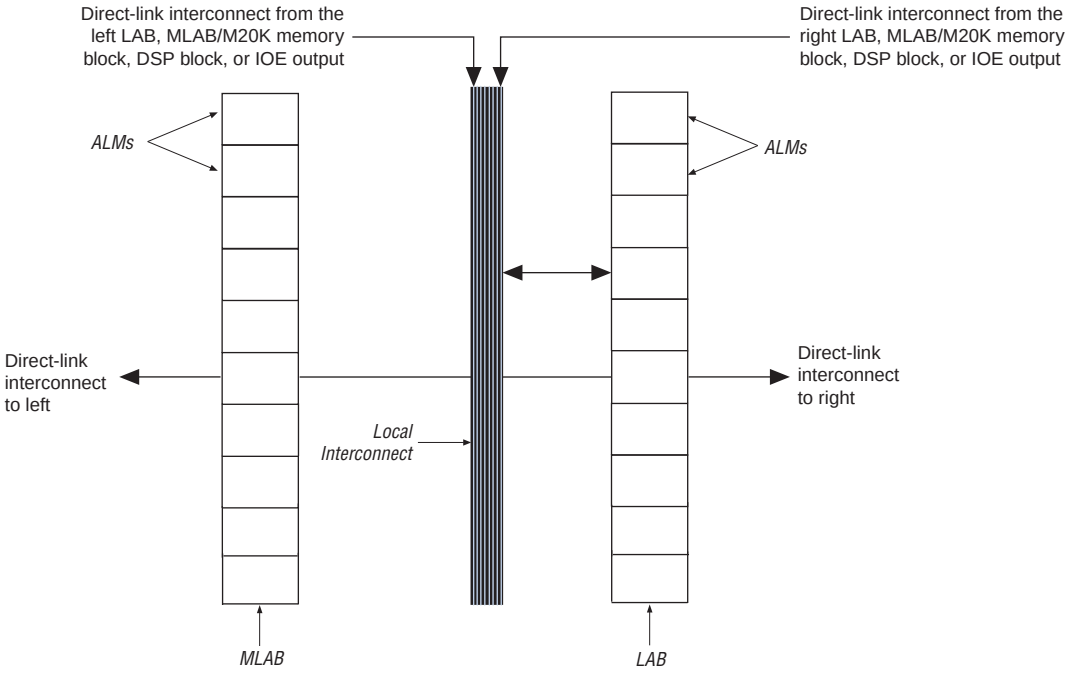
(1) You can use the MLAB ALM as a regular LAB ALM or configure it as a dual-port SRAM.

LAB Interconnects

The LAB local interconnect can drive ALMs in the same LAB. It is driven by column and row interconnects and ALM outputs in the same LAB. Neighboring LABs/MLABs, M20K blocks, or digital signal processing (DSP) blocks from the left or right can also drive the LAB's local interconnect through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility. Each LAB can drive 30 ALMs through fast-local and direct-link interconnects.

Figure 1–3 shows the direct-link connection.

Figure 1–3. Direct-Link Connection


Direct-link interconnect from the left LAB, MLAB/M20K memory block, DSP block, or IOE output
Direct-link interconnect from the right LAB, MLAB/M20K memory block, DSP block, or IOE output
ALMs
ALMs
Direct-link interconnect to left
Direct-link interconnect to right
Local Interconnect
MLAB
LAB

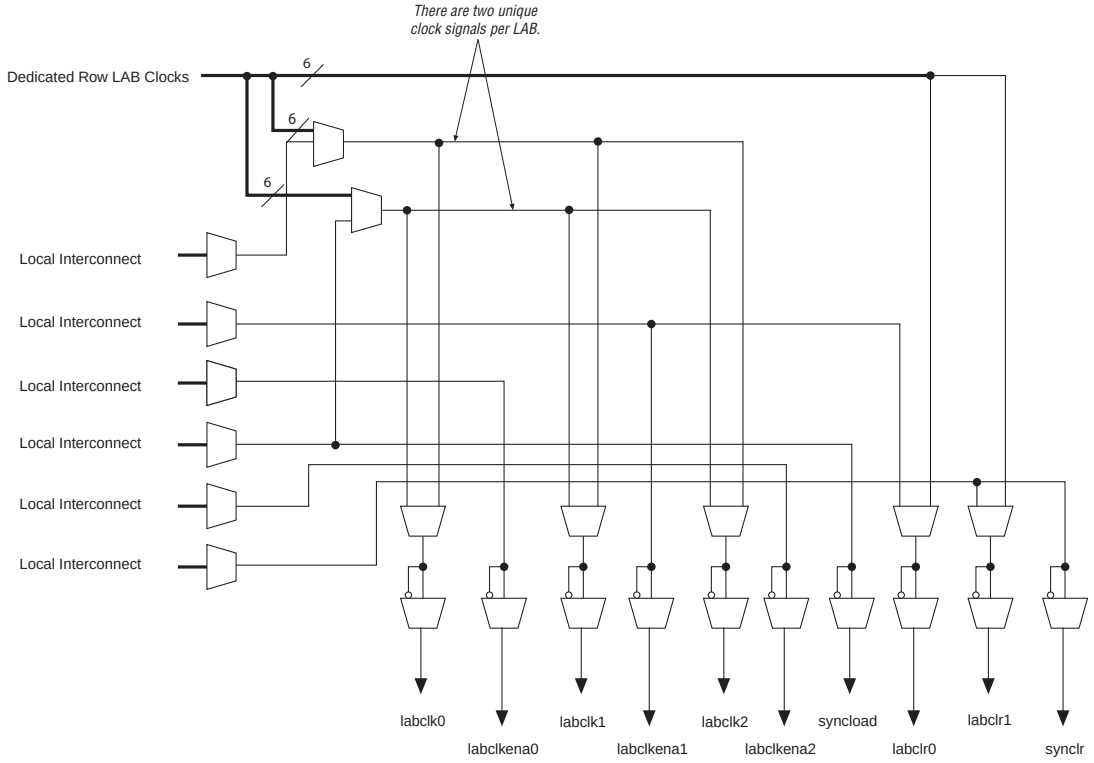
LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its ALMs. The control signals include three clocks, three clock enables, two asynchronous clears, one synchronous clear, and one synchronous load, for a maximum of 10 control signals at a time. Although you generally use synchronous load and clear signals when implementing counters, you can also use them with other functions.

Each LAB has two unique clock sources and three clock enable signals, as shown in Figure 1–4. The LAB control block can generate up to three clocks using two clock sources and three clock enable signals. Each LAB's clock and clock enable signals are linked. For example, any ALM in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. If the LAB uses both the rising and falling edges of a clock, it also uses two LAB-wide clock signals. Deasserting the clock enable signal turns off the corresponding LAB-wide clock.

The LAB row clocks [5..0] and LAB local interconnects generate the LAB-wide control signals. The MultiTrack interconnect's inherent low skew allows clock and control signal distribution in addition to data. The MultiTrack interconnect consists of continuous, performance-optimized routing lines of different lengths and speeds used for inter- and intra-design block connectivity.

Figure 1–4. LAB-Wide Control Signals



Adaptive Logic Modules

The ALM is the basic building block of logic in the Stratix V architecture. It provides advanced features with efficient logic utilization. Each ALM contains a variety of LUT-based resources that can be divided between two combinational adaptive LUTs (ALUTs) and four registers. With up to eight inputs for the two combinational ALUTs, one ALM can implement various combinations of two functions. This adaptability allows an ALM to be completely backward-compatible with four-input LUT architectures. One ALM can also implement any function with up to six inputs and certain seven-input functions.

In addition to the adaptive LUT-based resources, each ALM contains four programmable registers, two dedicated full adders, a carry chain, and a shared arithmetic chain. Through these dedicated resources, an ALM can efficiently implement various arithmetic functions and shift registers. Each ALM drives all types of interconnects: local, row, column, carry chain, shared arithmetic chain, and direct link. Figure 1–5 shows a high-level block diagram of the Stratix V ALM.

Figure 1–5. High-Level Block Diagram of the Stratix V ALM

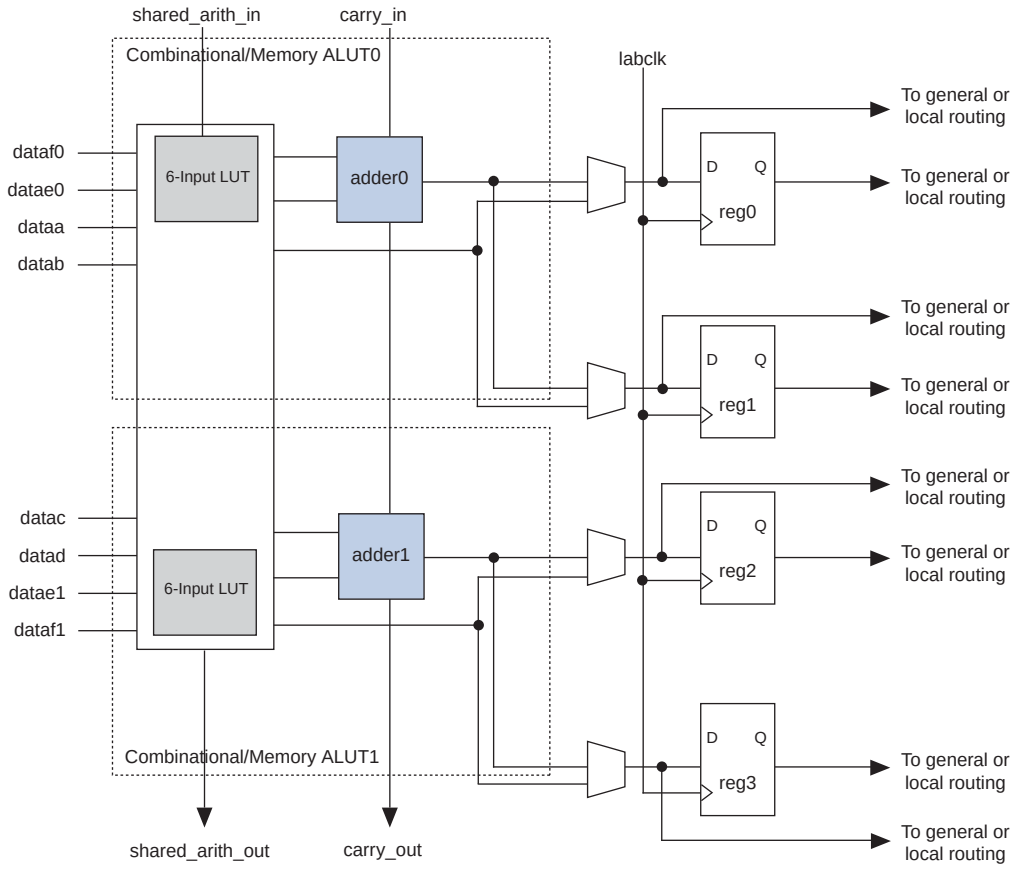
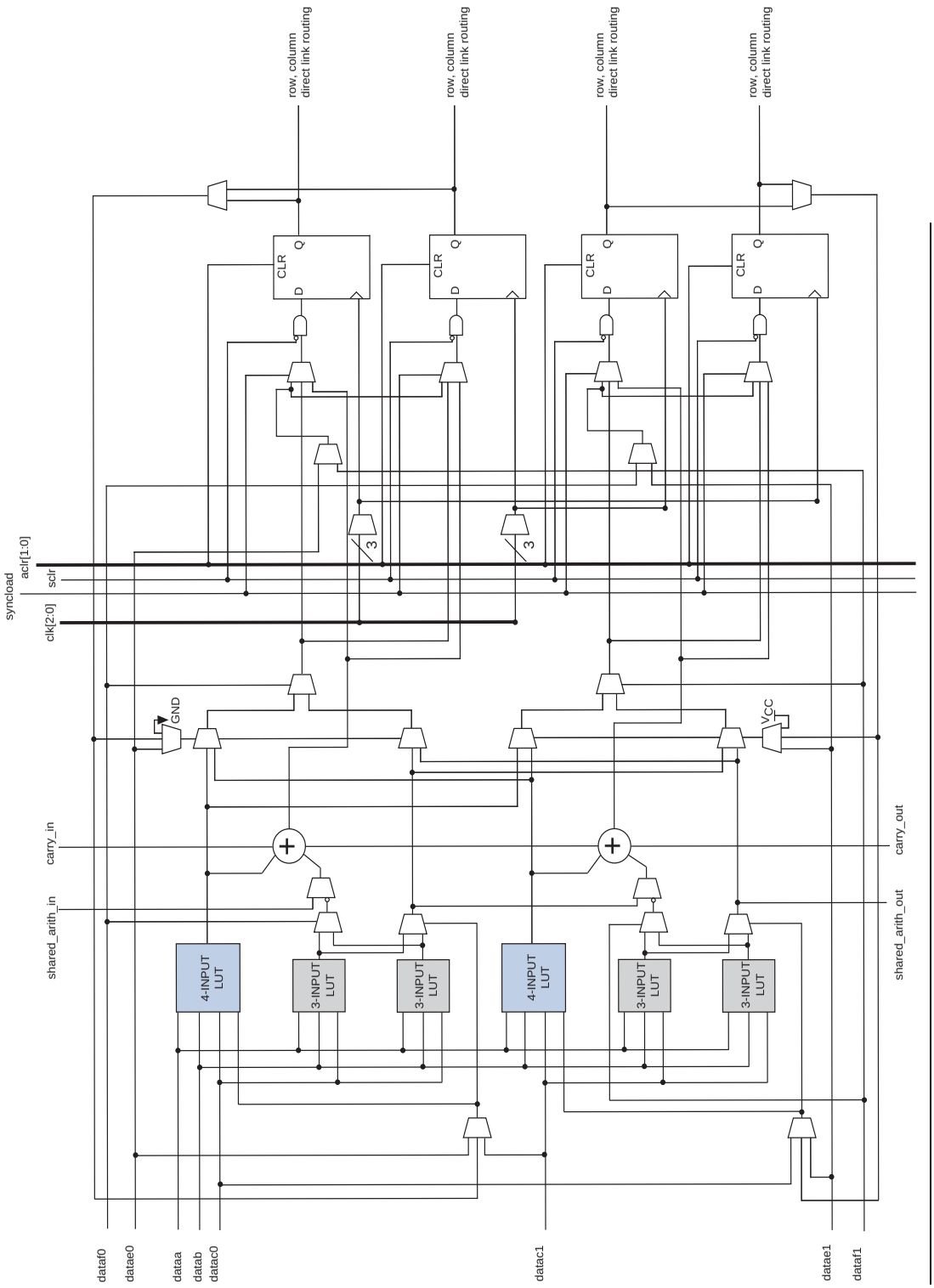


Figure 1–6 shows a detailed view of all the connections in an ALM.

Figure 1–6. ALM Connection Details for Stratix V Devices



One ALM contains four programmable registers. Each register has data, clock, synchronous and asynchronous clear, and synchronous load. Global signals, general-purpose I/O pins, or any internal logic can drive the register's clock and clear-control signals. Either general-purpose I/O pins or internal logic can drive the clock enable. For combinational functions, the register is bypassed and the output of the LUT drives directly to the outputs of an ALM.

Each ALM has two sets of outputs that drive the local, row, and column routing resources. The LUT, adder, or register outputs can drive these output drivers (refer to Figure 1–6). For each set of output drivers, two ALM outputs can drive column, row, or direct-link routing connections. One of these ALM outputs can also drive local interconnect resources. This allows the LUT or adder to drive one output while the register drives another output.

This feature, called register packing, improves device utilization because the device can use the register and the combinational logic for unrelated functions. Another special packing mode allows the register output to feed back into the LUT of the same ALM so that the register is packed with its own fan-out LUT. This provides another mechanism for improved fitting. The ALM can also drive out registered and unregistered versions of the LUT or adder output.

ALM Operating Modes

The Stratix V ALM operates in one of the following modes:

- "Normal Mode" on page 1–7
- "Extended LUT Mode" on page 1–9
- "Arithmetic Mode" on page 1–10
- "Shared Arithmetic Mode" on page 1–11

Each mode uses ALM resources differently. In each mode, eleven available inputs to an ALM—the eight data inputs from the LAB local interconnect, carry-in from the previous ALM or LAB, and the shared arithmetic chain connection from the previous ALM or LAB—are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear, synchronous clear, synchronous load, and clock enable control for the register. These LAB-wide signals are available in all ALM modes.

For more information about the LAB-wide control signals, refer to "LAB Control Signals" on page 1–3.

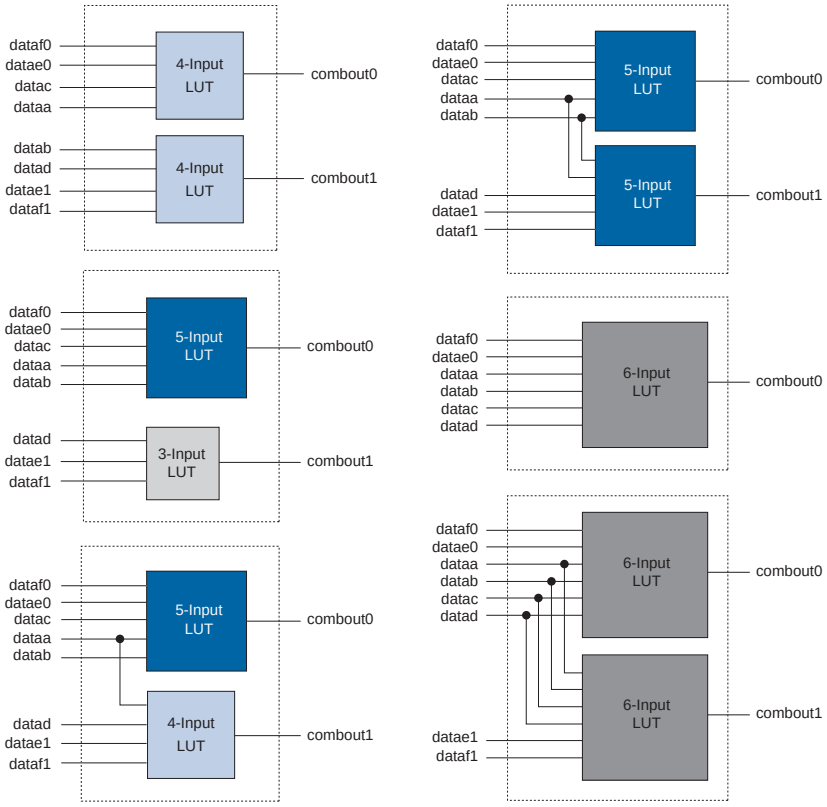
The Quartus II software and supported third-party synthesis tools, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions, automatically choose the appropriate mode for common functions such as counters, adders, subtractors, and arithmetic functions.

Normal Mode

Normal mode is suitable for general logic applications and combinational functions. In this mode, up to eight data inputs from the LAB local interconnect are inputs to the combinational logic. Normal mode allows two functions to be implemented in one Stratix V ALM, or a single function of up to six inputs. The ALM can support certain combinations of completely independent functions and various combinations of functions that have common inputs.

Figure 1–7 shows the supported LUT combinations in normal mode.

Figure 1–7. ALM in Normal Mode *(1)*



Note to Figure 1–7:

(1) Combinations of functions with fewer inputs than those shown are also supported. For example, combinations of functions with the following number of inputs are supported: 4 and 3, 3 and 3, 3 and 2, and 5 and 2.

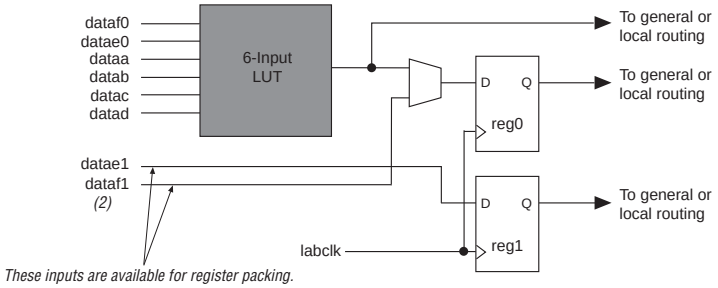
Normal mode provides complete backward-compatibility with four-input LUT architectures.

For the packing of 2 five-input functions into one ALM, the functions must have at least two common inputs. The common inputs are `dataa` and `datab`. The combination of a four-input function with a five-input function requires one common input (either `dataa` or `datab`).

In the case of implementing 2 six-input functions in one ALM, four inputs must be shared and the combinational function must be the same. In a sparsely used device, functions that could be placed in one ALM may be implemented in separate ALMs by the Quartus II software to achieve the best possible performance. As a device begins to fill up, the Quartus II software automatically uses the full potential of the Stratix V ALM. The Quartus II Compiler automatically searches for functions using common inputs or completely independent functions to be placed in one ALM to make efficient use of device resources. In addition, you can manually control resource use by setting location assignments.

You can implement any six-input function using inputs `dataa`, `datab`, `datac`, `datad`, and either `datae0` and `dataf0` or `datae1` and `dataf1`. If you use `datae0` and `dataf0`, the output is either driven to `register0`, `register0` is bypassed, or the output driven to `register0` and `register0` is bypassed, and the data drives out to the interconnect using the top set of output drivers (refer to Figure 1–8). If you use `datae1` and `dataf1`, the output either drives to `register1` or bypasses `register1` and drives to the interconnect using the bottom set of output drivers. The Quartus II Compiler automatically selects the inputs to the LUT. ALMs in normal mode support register packing.

Figure 1–8. Input Function in Normal Mode *(1)*



Notes to Figure 1–8:

(1) If you use `datae1` and `dataf1` as inputs to a six-input function, `datae0` and `dataf0` are available for register packing.

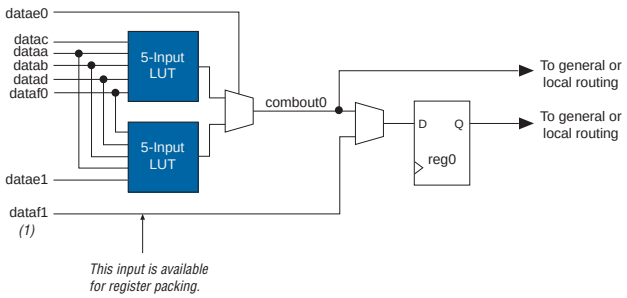
(2) The `dataf1` input is available for register packing only if the six-input function is unregistered.

Extended LUT Mode

Use extended LUT mode to implement a specific set of seven-input functions. The set must be a 2-to-1 multiplexer fed by two arbitrary five-input functions sharing four inputs. Figure 1–9 shows the template of supported seven-input functions using extended LUT mode. In this mode, if the seven-input function is unregistered, the unused eighth input is available for register packing.

Functions that fit into the template shown in Figure 1–9 occur naturally in designs. These functions often appear in designs as "if-else" statements in Verilog HDL or VHDL code.

Figure 1–9. Template for Supported Seven-Input Functions in Extended LUT Mode



Note to Figure 1–9:

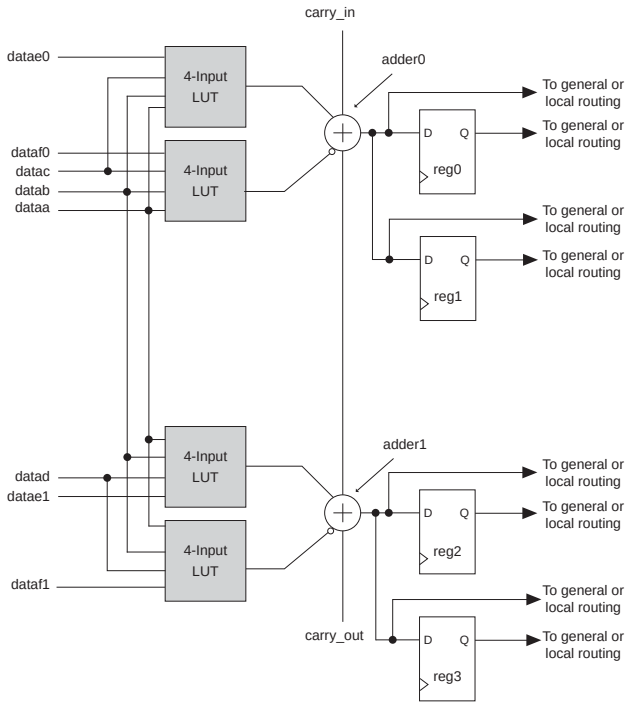
(1) If the seven-input function is unregistered, the unused eighth input is available for register packing. The second register, `reg1`, is not available.

Arithmetic Mode

Arithmetic mode is ideal for implementing adders, counters, accumulators, wide parity functions, and comparators. The ALM in arithmetic mode uses two sets of 2 four-input LUTs along with two dedicated full adders. The dedicated adders allow the LUTs to be available to perform pre-adder logic; therefore, each adder can add the output of 2 four-input functions.

The four LUTs share `dataa` and `datab` inputs. As shown in Figure 1–10, the carry-in signal feeds to `adder0` and the carry-out from `adder0` feeds to the carry-in of `adder1`. The carry-out from `adder1` drives to `adder0` of the next ALM in the LAB. ALMs in arithmetic mode can drive out either registered, unregistered, or registered and unregistered versions of the adder outputs.

Figure 1–10. ALM in Arithmetic Mode



While operating in arithmetic mode, the ALM can support simultaneous use of the adder's carry output along with combinational logic outputs. In this operation, adder output is ignored. Using the adder with combinational logic output provides resource savings of up to 50% for functions that can use this ability.

Arithmetic mode also offers clock enable, counter enable, synchronous up/down control, add/subtract control, synchronous clear, and synchronous load. The LAB local interconnect data inputs generate the clock enable, counter enable, synchronous up/down, and add/subtract control signals. These control signals are good candidates for the inputs that are shared between the four LUTs in the ALM. The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. You can individually disable or enable these signals for each register. The Quartus II software automatically places any registers that are not used by the counter into other LABs.

Carry Chain

The carry chain provides a fast carry function between the dedicated adders in arithmetic or shared-arithmetic mode. The two-bit carry select feature in Stratix V devices halves the propagation delay of carry chains within the ALM. Carry chains can begin in either the first ALM or the fifth ALM in the LAB. The final carry-out signal is routed to the ALM, where it is fed to local, row, or column interconnects.

The Quartus II Compiler automatically creates carry-chain logic during design processing, or you can create it manually during design entry. Parameterized functions such as LPM functions automatically take advantage of carry chains for the appropriate functions.

The Quartus II Compiler creates carry chains longer than 20 (10 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically, allowing fast horizontal connections to MLAB/M20K memory and DSP blocks. A carry chain can continue as far as a full column.

To avoid routing congestion in one small area of the device when a high fan-in arithmetic function is implemented, the LAB can support carry chains that only use either the top half or bottom half of the LAB before connecting to the next LAB. This leaves the other half of the ALMs in the LAB available for implementing narrower fan-in functions in normal mode. Carry chains that use the top five ALMs in the first LAB carry into the top half of the ALMs in the next LAB within the column. Carry chains that use the bottom five ALMs in the first LAB carry into the bottom half of the ALMs in the next LAB within the column. In every alternate LAB column, the top half can be bypassed; in the other MLAB columns, the bottom half can be bypassed.

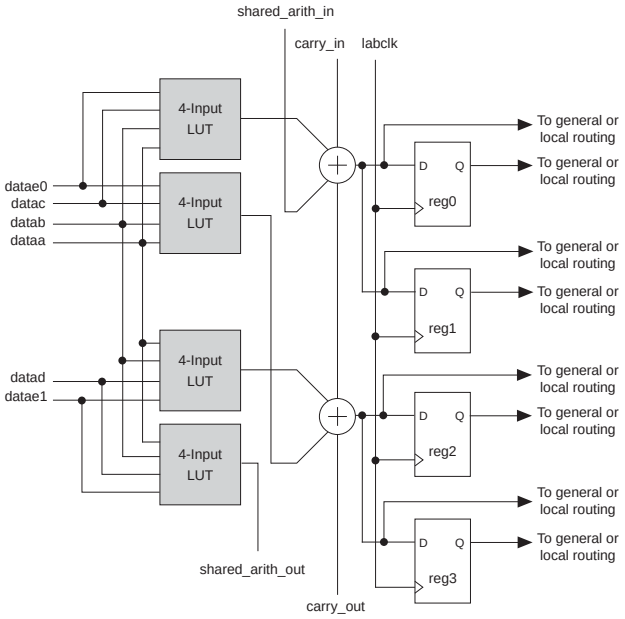
For more information about carry-chain interconnects, refer to "ALM Interconnects" on page 1–13.

Shared Arithmetic Mode

In shared arithmetic mode, the ALM can implement a three-input add within the ALM. In this mode, the ALM is configured with 4 four-input LUTs. Each LUT either computes the sum of three inputs or the carry of three inputs. The output of the carry computation is fed to the next adder (either to `adder1` in the same ALM or to `adder0` of the next ALM in the LAB) using a dedicated connection called the shared arithmetic chain. This shared arithmetic chain can significantly improve the performance of an adder tree by reducing the number of summation stages required to implement an adder tree.

Figure 1–11 shows the ALM using this feature.

Figure 1–11. ALM in Shared Arithmetic Mode



You can find adder trees in many different applications. For example, you can implement the summation of the partial products in a logic-based multiplier in a tree structure. Another example is a correlator function that can use a large adder tree to sum filtered data samples in a given time frame to recover or de-spread data that was transmitted using spread-spectrum technology.

Shared Arithmetic Chain

The shared arithmetic chain available in enhanced arithmetic mode allows the ALM to implement a three-input add. This significantly reduces the resources necessary to implement large adder trees or correlator functions.

The shared arithmetic chains can begin in either the first or sixth ALM in the LAB. The Quartus II Compiler creates shared arithmetic chains longer than 20 (10 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long shared arithmetic chain runs vertically, allowing fast horizontal connections to the MLAB/M20K memory and DSP blocks. A shared arithmetic chain can continue as far as a full column.

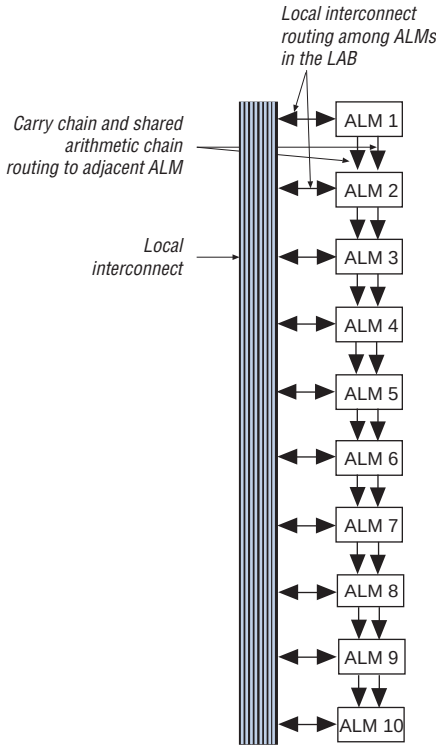
Similar to carry chains, the top and bottom halves of shared arithmetic chains in alternate LAB columns can be bypassed. This capability allows the shared arithmetic chain to cascade through half of the ALMs in a LAB while leaving the other half available for narrower fan-in functionality. The top half of every other LAB column can be bypassed, while the bottom half of the other LAB columns can be bypassed.

For more information on shared arithmetic chain interconnect, refer to "ALM Interconnects" on page 1–13.

ALM Interconnects

There are two dedicated paths between ALMs—carry chain and shared arithmetic chain. Stratix V devices include an enhanced interconnect structure in LABs for routing shared arithmetic chains and carry chains for efficient arithmetic functions. These ALM-to-ALM connections bypass the local interconnect. The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance. Figure 1–12 shows the shared arithmetic chain and carry chain interconnects.

Figure 1–12. Shared Arithmetic Chain and Carry Chain Interconnects



Clear and Preset Logic Control

LAB-wide signals control the logic for the register's clear signal. The ALM directly supports an asynchronous clear function. You can achieve the register preset through the Quartus II software's **NOT-gate push-back logic** option. Each LAB supports up to two clears.

Stratix V devices provide a device-wide reset pin (DEV_CLRn) that resets all the registers in the device. An option set before compilation in the Quartus II software controls this pin. This device-wide reset overrides all other control signals.

LAB Power Management Techniques

The following techniques are used to manage static and dynamic power consumption within the LAB:

- To save AC power, the Quartus II software forces all adder inputs low when the ALM adders are not in use.
- Stratix V LABs operate in high-performance mode or low-power mode. The Quartus II software automatically chooses the appropriate mode for the LAB, based on your design and to optimize speed versus leakage trade-offs.
- Clocks represent a significant portion of dynamic power consumption because of their high switching activity and long paths. The LAB clock that distributes a clock signal to registers within a LAB is a significant contributor to overall clock power consumption. Each LAB's clock and clock enable signals are linked. For example, a combinational ALUT or register in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. To disable a LAB-wide clock power consumption without disabling the entire clock tree, use the LAB-wide clock enable to gate the LAB-wide clock. The Quartus II software automatically promotes register-level clock enable signals to the LAB-level. All registers within the LAB that share a common clock and clock enable are controlled by a shared, gated clock. To take advantage of these clock enables, use a clock-enable construct in your HDL code for the registered logic.

For more information about implementing static and dynamic power consumption within the LAB, refer to the *Power Optimization* chapter in volume 2 of the *Quartus II Handbook*.

Document Revision History

Table 1–1 lists the revision history for this chapter.

Table 1–1. Document Revision History

Date	Version	Changes
June 2012	1.4	■ Updated Figure 1–5, Figure 1–6, and Figure 1–12. ■ Removed register chain expression. ■ Minor text edits.
November 2011	1.3	■ Updated Figure 1–1, Figure 1–4, and Figure 1–6. ■ Removed "Register Chain" section.
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Figure 1–6. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

2. Memory Blocks in Stratix V Devices

SV51003-1.4

This chapter describes the embedded memory blocks in Stratix® V devices.

Embedded memory blocks provide different sizes of embedded SRAM to address the Stratix V device design requirements efficiently. Embedded memory blocks include 640-bit enhanced memory logic array blocks (MLABs) and 20-Kbit M20K blocks. MLABs are optimized to implement shift registers for digital signal processing (DSP) applications, wide shallow FIFO buffers, and filter delay lines. You can use the M20K blocks to support larger memory configurations and include error correction code (ECC).

This chapter contains the following sections:

- "Overview" on page 2–1
- "Memory Modes" on page 2–10
- "Clocking Modes" on page 2–17
- "Design Considerations" on page 2–18

Overview

Table 2–1 lists the features supported by the embedded memory blocks.

Table 2–1. Summary of Memory Features in Stratix V Devices (Part 1 of 2)

Feature	MLABs	M20K
Maximum performance	600 MHz	600 MHz
Total RAM bits (including parity bits)	640	20,480
Configurations (depth x width)	64 x 8 64 x 9 64 x 10 32 x 16 32 x 18 32 x 20	16K x 1 8K x 2 4K x 4 4K x 5 2K x 8 2K x 10 1K x 16 1K x 20 512 x 32 512 x 40
Parity bits	Y	Y
Byte enable	Y	Y

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Table 2–1. Summary of Memory Features in Stratix V Devices (Part 2 of 2)

Feature	MLABs	M20K
Packed mode	—	Y
Address clock enable	Y	Y
Single-port memory	Y	Y
Simple dual-port memory	Y	Y
True dual-port memory	—	Y
Embedded shift register	Y	Y
ROM	Y	Y
FIFO buffer	Y	Y
Simple dual-port mixed width support	—	Y
True dual-port mixed width support	—	Y
Memory Initialization File (**.mif**)	Y	Y
Mixed-clock mode	Y	Y
Power-up condition	Outputs cleared if registered, otherwise reads memory contents	Outputs cleared
Register clears	Output registers	Output registers
Write/Read operation triggering	Write and Read—Rising clock edges	Write and Read—Rising clock edges
Same-port read-during-write	Outputs set to **don't care**	Outputs set to **new data**
Mixed-port read-during-write *(1)*	Outputs set to **new data**, **old data**, **don't care**, or **constrained don't care** *(2)*	Outputs set to **old data** or **don't care**
ECC support	Soft IP support using the Quartus® II software	Built-in support in x32-wide simple dual-port mode or soft IP support using the Quartus II software

Notes to Table 2–1:

(1) You must use the same clock for both read and write ports. For more information, refer to "Mixed-Port Read-During-Write Mode" on page 2–20.

(2) When you enable output register in the RAM MegaWizard™ Plug-In Manager in the Quartus II software, you can set the outputs to **new data**, **old data**, **don't care**, or **constrained don't care**.

Table 2–2 lists the capacity and distribution of the embedded memory blocks in each Stratix V device.

Table 2–2. Memory Capacity and Distribution in Stratix V Devices

Family	Device	MLABs	M20K Blocks	Total Dedicated RAM Bits (M20K Blocks Only) (Mb)	Total RAM Bits (Including LABs) (Mb)
Stratix V GX	5SGXA3	6,415	957	19	22.92
	5SGXA4	7,925	1,900	37	41.84
	5SGXA5	9,250	2,304	45	50.65
	5SGXA7	11,736	2,560	50	57.16
	5SGXA9	15,850	2,640	52	61.67
	5SGXAB	17,960	2,640	52	62.96
	5SGXB5	9,250	2,100	41	46.65
	5SGXB6	11,270	2,660	52	58.88
	5SGXB9	15,850	2,640	52	61.67
	5SGXBB	17,960	2,640	52	62.96
Stratix V GT	5SGTC5	8,020	2,304	45	49.90
	5SGTC7	11,736	2,560	50	57.16
Stratix V GS	5SGSD3	4,450	688	13	15.72
	5SGSD4	6,792	957	19	23.15
	5SGSD5	8,630	2,014	39	44.27
	5SGSD6	11,000	2,320	45	51.71
	5SGSD8	13,120	2,567	50	58.01
Stratix V E	5SEE9	15,850	2,640	52	61.67
	5SEEB	17,960	2,640	52	62.96

Embedded Memory Block Types

M20K memory blocks are dedicated resources. MLABs are dual-purpose blocks. You can configure the MLABs as regular logic array blocks (LABs) or as MLABs. Ten adaptive logic modules (ALMs) make up one MLAB. You can configure each ALM in an MLAB as either a 64 x 1 or a 32 x 2 block, resulting in a 64 x 10 or a 32 x 20 simple dual-port SRAM block in a single MLAB.

Parity Bit Support

On MLABs, the ninth bit associated with each byte can store a parity bit or serve as an additional data bit. No parity function is actually performed on the ninth bit.

The M20K supports one parity bit per 4-data bits when the data width is 5, 10, 20, or 40. The parity bits for inputs and outputs are bit 4, 9, 14, 19, 24, 29, 34, and 39. When writing or reading with non-parity widths, these bits are skipped. No parity function is performed on bit 4, 9, 14, 19, 24, 29, 34, and 39.

Byte Enable Support

All embedded memory blocks support byte enables that mask the input data so that only specific bytes of data are written. The unwritten bytes retain the previously written values. The write enable (`wren`) signals, along with the byte enable (`byteena`) signals, control the write operations of the RAM blocks.

The default value for the byte enable signals is high (enabled), in which case writing is controlled only by the write enable signals. The byte enable registers do not have a clear port. When using parity bits on the M20K blocks, the byte enable controls ten bits (eight bits of data plus two parity bits). When using parity bits on the MLAB, the byte enable controls all ten bits in the widest mode. Byte enables operate in a one-hot fashion, with the LSB of the `byteena` signal corresponding to the LSB of the data bus. The byte enables are active high.

Table 2–3 lists the `byteena` controls in the x40 data width.

Table 2–3. byteena Controls in x40 Data Width

byteena[3..0]	Data Bits Written			
1111(default)	[39:30]	[29:20]	[19:10]	[9:0]
1000	[39:30]	—	—	—
0100	—	[29:20]	—	—
0010	—	—	[19:10]	—
0001	—	—	—	[9:0]

Table 2–4 lists the `byteena` controls in the x20 data width.

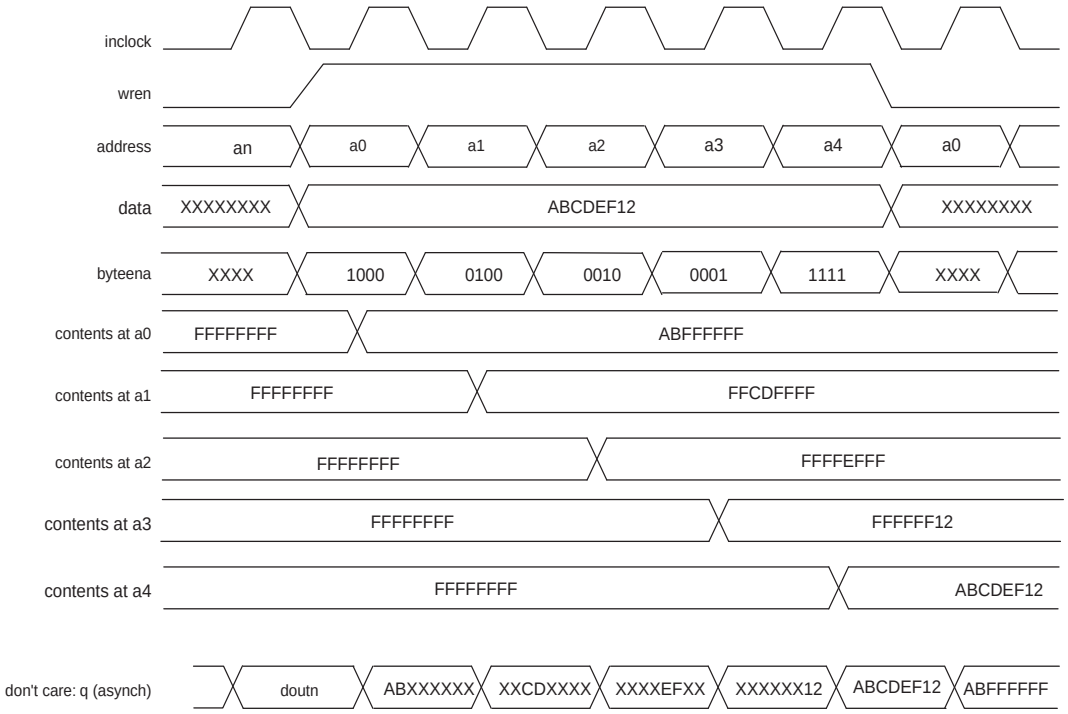
Table 2–4. byteena Controls in x20 Data Width

byteena[1:0]	Data Bits Written	
11(default)	[19:10]	[9:0]
10	[19:10]	—
01	—	[9:0]

If you use the ECC feature on the M20K blocks, you cannot use the byte enable feature.

Figure 2–1 shows how the wren and byteena signals control the operations of the RAM blocks. When a byte-enable bit is deasserted during a write cycle, the corresponding data byte output can appear as either a "don't care" value or the current data at that location. The output value for the masked byte is controllable with the Quartus II software. When a byte-enable bit is asserted during a write cycle, the corresponding data byte output also depends on the setting chosen in the Quartus II software.

Figure 2–1. Byte Enable Functional Waveform



Packed Mode Support

Stratix V M20K memory blocks support packed mode. The packed mode feature packs two independent single-port RAMs into one memory block. The Quartus II software automatically implements packed mode where appropriate by placing the physical RAM block in true dual-port mode and using the MSB of the address to distinguish between the two logical RAMs. The size of each independent single-port RAM must not exceed half of the target block size.

Address Clock Enable Support

Stratix V embedded memory blocks support address clock enable, which holds the previous address value for as long as the signal is enabled (addressstall = 1). When the memory blocks are configured in dual-port mode, each port has its own independent address clock enable. The default value for the address clock enable signal is low (disabled).

Figure 2–2 shows an address clock enable block diagram. The address clock enable is referred to by the port name `addressstall`.

Figure 2–2. Address Clock Enable

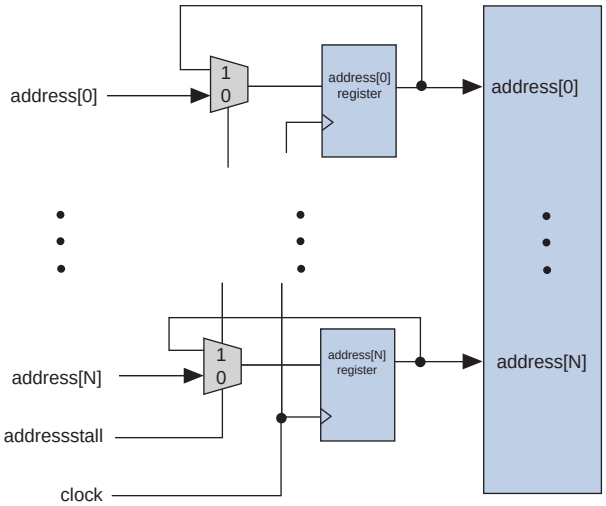


Figure 2–3 shows the address clock enable waveform during the read cycle.

Figure 2–3. Address Clock Enable During Read Cycle Waveform

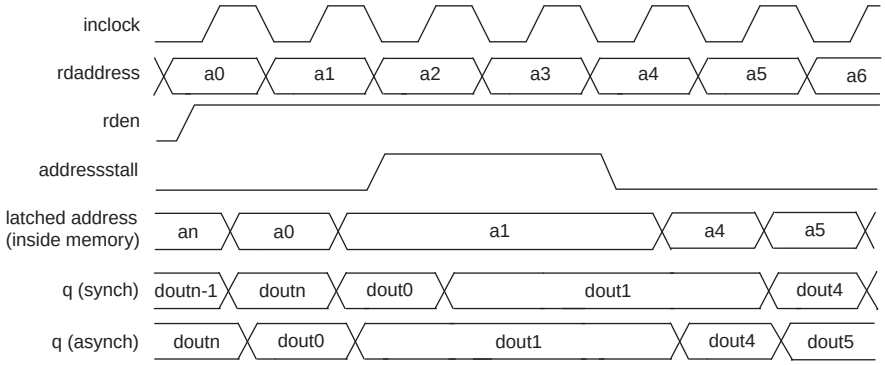
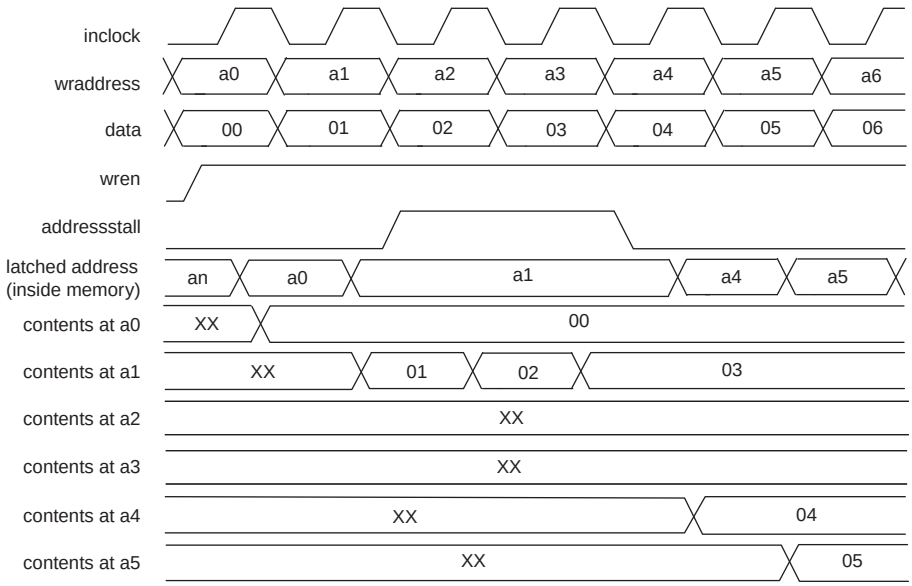


Figure 2–4 shows the address clock enable waveform during the write cycle.

Figure 2–4. Address Clock Enable During the Write Cycle Waveform



Mixed Width Support

M20K memory blocks support mixed data widths inherently. MLABs can support mixed data widths through emulation with the Quartus II software. When using simple dual-port, true dual-port, or FIFO modes, mixed width support allows you to read and write different data widths to a memory block. For more information about the different widths supported per memory mode, refer to "Memory Modes" on page 2–10.

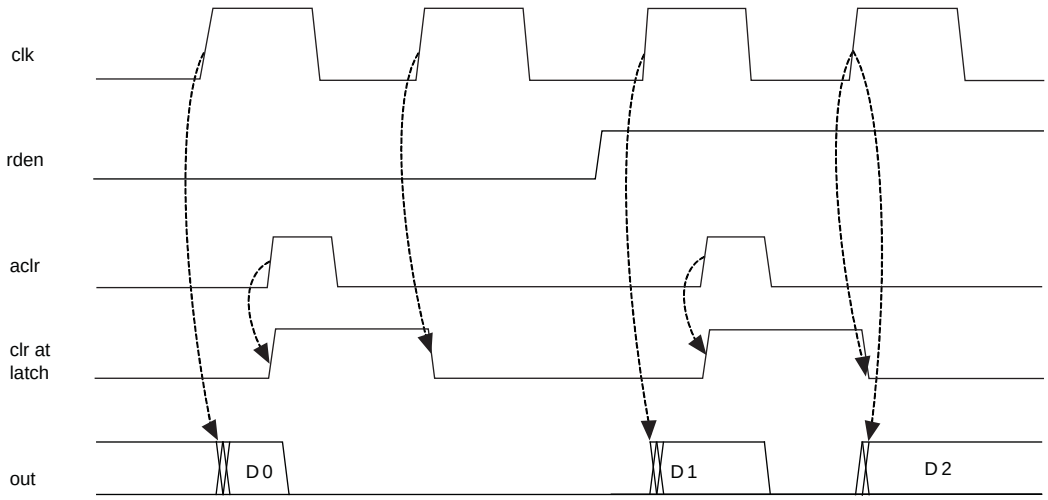
MLABs do not support mixed-width FIFO mode.

Asynchronous Clear

M20K memory blocks support asynchronous clear on output latches and output registers. Therefore, if your RAM does not use output registers, clear the RAM outputs using the output latch asynchronous clear. Because the clear is an asynchronous signal, it is generated at any time. The internal logic extends the clear pulse until the next rising edge of the output clock. When the clear is asserted, the outputs are cleared and stay cleared until the next read cycle.

Figure 2–5 shows the output latch clear in Stratix V devices.

Figure 2–5. Output Latch Clear in Stratix V Devices



Error Correction Code Support

M20K blocks have built-in support for ECC when in x32-wide simple dual-port mode. ECC allows you to detect and correct data errors at the output of the memory. ECC can perform single-error correction, double-adjacent-error correction, and triple-adjacent-error detection in a 32-bit word; however, ECC cannot detect four or more errors.

The M20K runs slower than non-ECC simple-dual port mode when ECC is engaged; however, you can enable optional ECC pipeline registers before the output decoder to achieve the same performance as non-ECC simple-dual port mode at the expense of one cycle of latency.

The M20K ECC status is communicated with two ECC status flag signals—`e` (error) and `ue` (uncorrectable error). The status flags are part of the regular output from the memory block. When ECC is engaged, you cannot access two of the parity bits because they are replaced by the ECC status flag.

Table 2–5 lists the truth table for the ECC status flags.

Table 2–5. Truth Table for the ECC Status Flags *(1)*

e (error)	ue (uncorrectable error)	Status
0	0	No error.
0	1	Illegal.
1	0	A correctable error occurred and the error has been corrected at the outputs; however, the memory array has not been updated.
1	1	An uncorrectable error occurred and uncorrectable data appears at the outputs.

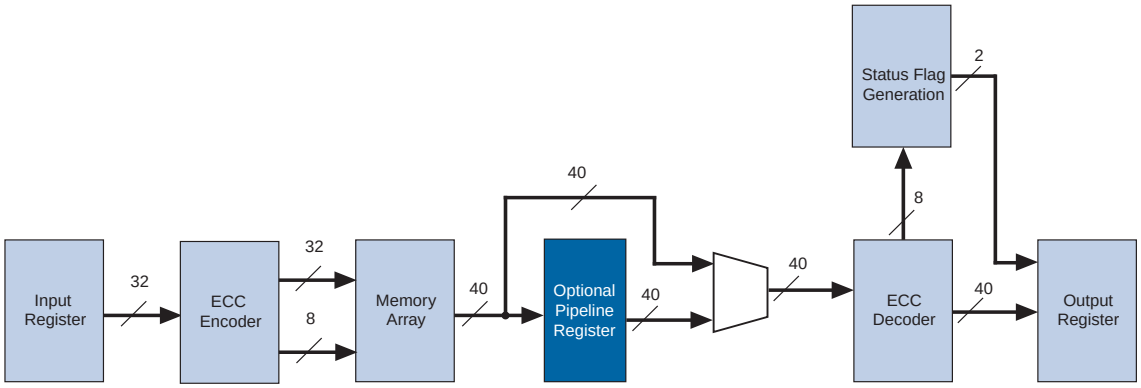
Note to Table 2–5:

(1) `eccstatus[1]` corresponds to `e` and `eccstatus[0]` corresponds to `ue`.

When ECC is engaged, you cannot use the byte enable feature. Also, when ECC is engaged, read-during-write old data mode is not supported.

Figure 2–6 shows a diagram of the ECC block for M20K.

Figure 2–6. ECC Block for M20K



Memory Modes

Stratix V embedded memory blocks allow you to implement fully synchronous SRAM memory in multiple modes of operation. M20K blocks do not support asynchronous memory (unregistered inputs). MLABs support asynchronous (flow-through) read operations.

Depending on which memory block you target, you can use the following modes:

- "Single-Port RAM" on page 2–10
- "Simple Dual-Port Mode" on page 2–12
- "True Dual-Port Mode" on page 2–14 (only supported on M20K)
- "Shift-Register Mode" on page 2–15
- "ROM Mode" on page 2–16
- "FIFO Mode" on page 2–16

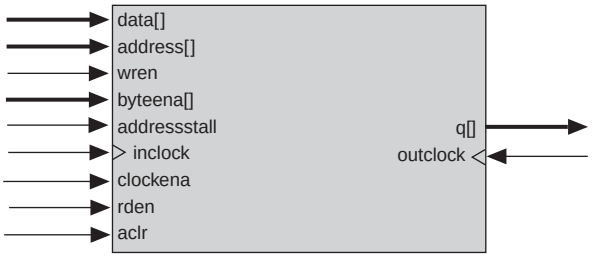
If you use the memory blocks in ROM, single-port, simple dual-port, or true dual-port mode, you can corrupt the memory contents if you violate the setup or hold-time on any of the memory block input registers. This applies to both read and write operations.

Single-Port RAM

All embedded memory blocks support single-port mode. Single-port mode allows you to do either one-read or one-write operation at a time.

Figure 2–7 shows the single-port RAM configuration.

Figure 2–7. Single-Port RAM *(1)*



Note to Figure 2–7:

(1) You can implement two single-port memory blocks in a single M20K block. For more information, refer to "Packed Mode Support" on page 2–5.

During a write operation, the RAM output behavior is configurable. If you use the read-enable signal and perform a write operation with the read enable deactivated, the RAM outputs retain the values they held during the most recent active read enable. If you activate read enable during a write operation or if you do not use the read-enable signal at all, the RAM outputs show the "new data" being written.

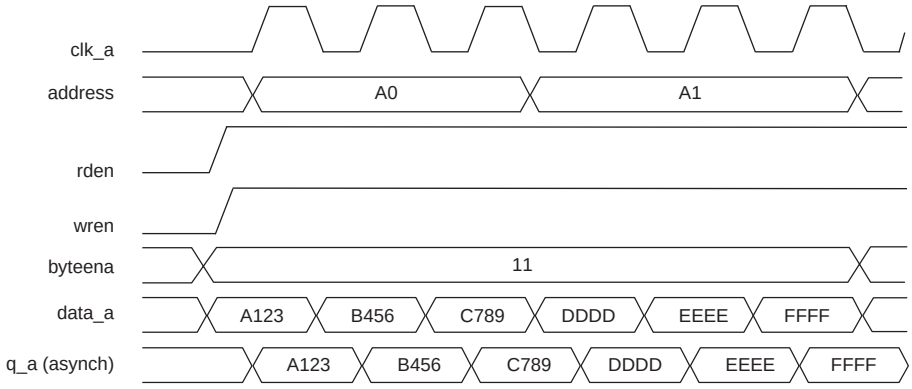
Table 2–6 lists the possible port width configurations for embedded memory blocks in single-port mode.

Table 2–6. Port Width Configurations for MLABs and M20K (Single-Port Mode)

Port Width Configurations	
MLABs	**M20K**
	16K x 1
	8K x 2
64 x 8	4K x 4
64 x 9	4K x 5
64 x 10	2K x 8
32 x 16	2K x 10
32 x 18	1K x 16
32 x 20	1K x 20
	512 x 32
	512 x 40

Figure 2–8 shows timing waveforms for read and write operations in single-port mode with unregistered outputs. Registering the RAM outputs delay the q output by one clock cycle.

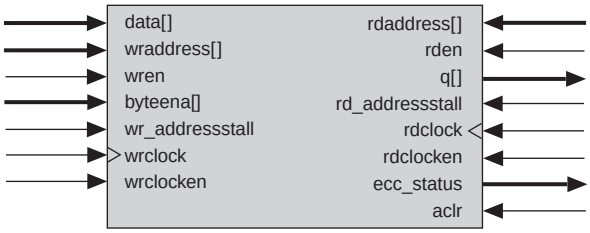
Figure 2–8. Timing Waveform for Read-Write Operations (Single-Port Mode)



Simple Dual-Port Mode

All embedded memory blocks support simple dual-port mode. Simple dual-port mode allows you to perform one-read and one-write operation to different locations at the same time. The write operation happens on port A; the read operation happens on port B. Figure 2–9 shows a simple dual-port configuration.

Figure 2–9. Simple Dual-Port Memory for Stratix V Devices *(1)*



Note to Figure 2–9:

(1) Simple dual-port RAM supports input/output clock mode and read/write clock mode.

Simple dual-port mode supports different read and write data widths (mixed-width support). Table 2–7 lists the mixed width configurations for M20K blocks in simple dual-port mode. MLABs do not have native support for mixed-width operations. The Quartus II software implements mixed-width memories in MLABs with more than one MLAB.

Table 2–7. M20K Block Mixed-Width Configurations (Simple Dual-Port Mode)

Read Port	Write Port									
	16K x 1	8K x 2	4K x 4	4K x 5	2K x 8	2K x 10	1K x 16	1K x 20	512 x 32	512 x 40
16K x 1	Y	Y	Y	—	Y	—	Y	—	Y	—
8K x 2	Y	Y	Y	—	Y	—	Y	—	Y	—
4K x 4	Y	Y	Y	—	Y	—	Y	—	Y	—
4K x 5	—	—	—	Y	—	Y	—	Y	—	Y
2K x 8	Y	Y	Y	—	Y	—	Y	—	Y	—
2K x 10	—	—	—	Y	—	Y	—	Y	—	Y
1K x 16	Y	Y	Y	—	Y	—	Y	—	Y	—
1K x 20	—	—	—	Y	—	Y	—	Y	—	Y
512 x 32	Y	Y	Y	—	Y	—	Y	—	Y	—
512 x 40	—	—	—	Y	—	Y	—	Y	—	Y

In simple dual-port mode, M20K blocks support separate write-enable and read-enable signals. You can save power by keeping the read-enable signal low (inactive) when not reading. Read-during-write operations to the same address can either output a "don't care" value or "old data" value. To choose the desired behavior, set the read-during-write behavior to either **don't care** or **old data** in the RAM MegaWizard Plug-In Manager. For more information, refer to "Read-During-Write Behavior" on page 2–19.

MLABs only support a write-enable signal. Read-during-write behavior for the MLABs can be a "new data", "don't care", "old data", or "constrained don't care" value. The available choices depend on the configuration of the MLAB.

Figure 2–10 shows timing waveforms for read and write operations in simple dual-port mode with unregistered outputs. Registering the RAM outputs delay the q output by one clock cycle.

Figure 2–10. Simple Dual-Port Timing Waveforms

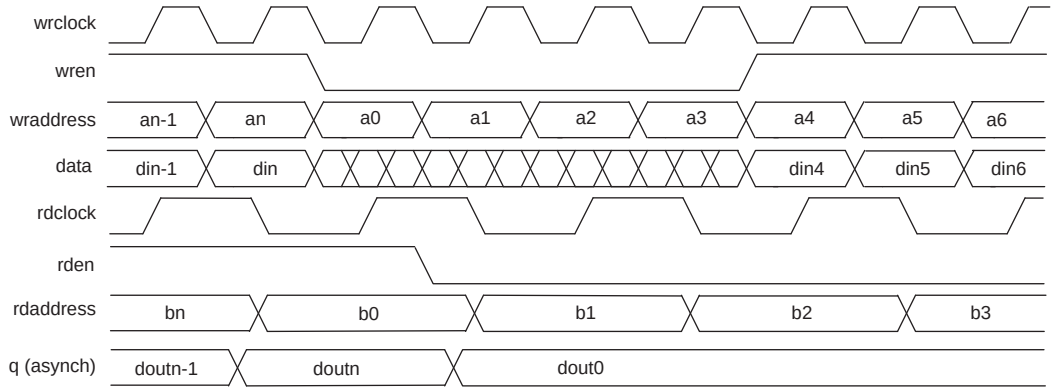
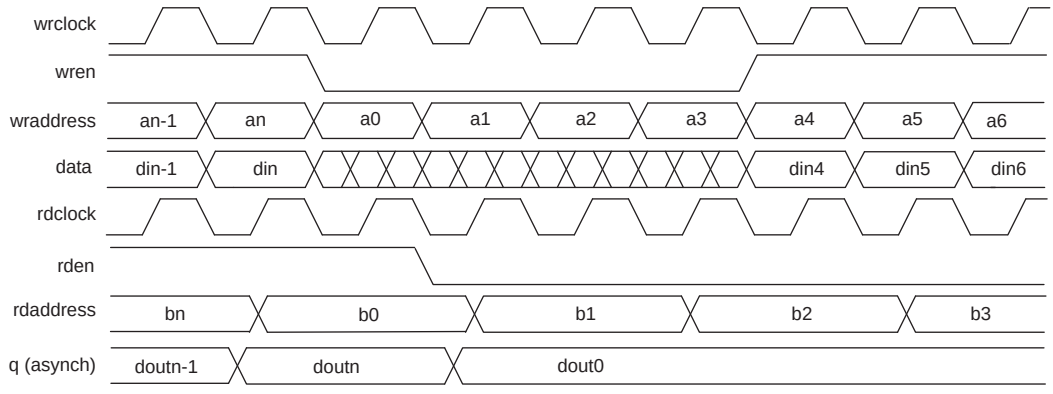


Figure 2–11 shows timing waveforms for read and write operations in mixed-port mode with unregistered outputs.

Figure 2–11. Mixed-Port Read-During-Write Timing Waveforms

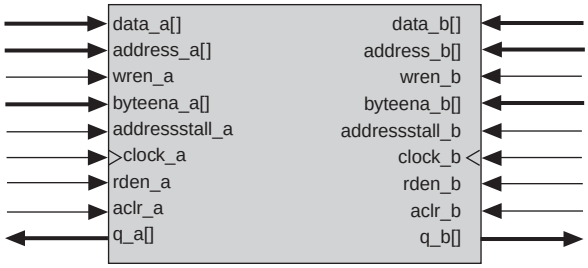


True Dual-Port Mode

Stratix V M20K blocks support true dual-port mode. Sometimes called bidirectional dual-port, this mode allows you to perform any combination of two port operations—two reads, two writes, or one read and one write at two different clock frequencies.

Figure 2–12 shows the true dual-port RAM configuration.

Figure 2–12. True-Dual Port Memory for Stratix V Devices



The widest bit configuration of the M20K blocks in true dual-port mode is 1k x 16-bit (x20-bit with parity).

Wider configurations are unavailable because the number of output drivers is equivalent to the maximum bit width of the respective memory block. Because true dual-port RAM has outputs on two ports, its maximum width equals half of the total number of output drivers.

Table 2–8 lists the possible M20K block mixed-port width configurations in true dual-port mode.

Table 2–8. M20K Block Mixed-Width Configurations (True Dual-Port Mode)

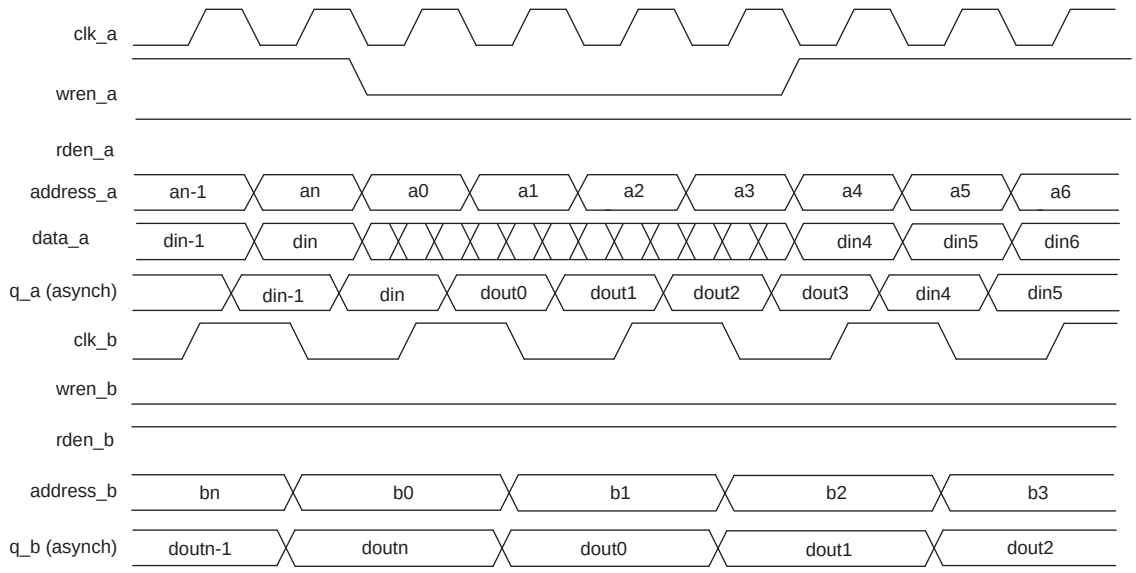
Port A	Port B							
	16K x 1	8K x 2	4K x 4	4K x 5	2K x 8	2K x 10	1K x 16	1K x 20
16K x 1	Y	Y	Y	—	Y	—	Y	—
8K x 2	Y	Y	Y	—	Y	—	Y	—
4K x 4	Y	Y	Y	—	Y	—	Y	—
4K x 5	—	—	—	Y	—	Y	—	Y
2K x 8	Y	Y	Y	—	Y	—	Y	—
2K x 10	—	—	—	Y	—	Y	—	Y
1K x 16	Y	Y	Y	—	Y	—	Y	—
1K x 20	—	—	—	Y	—	Y	—	Y

In true dual-port mode, M20K memory blocks support separate write-enable and read-enable signals. You can save power by keeping the read-enable signal low (inactive) when not reading. Read-during-write operations to the same address will give you the "new data" output at that location. Set the read-during-write behavior to **new data** in the RAM MegaWizard Plug-In Manager in the Quartus II software. For more information, refer to "Read-During-Write Behavior" on page 2–19.

In true dual-port mode, you can access any memory location at any time from either port. If you are accessing the same memory location from both ports, you must avoid possible write conflicts. A write conflict happens if you are writing to the same address location from both ports at the same time. This results in unknown data being stored to that address location. No conflict resolution circuitry is built into the Stratix V embedded memory blocks. You must resolve address conflicts external to the RAM block.

Figure 2–13 shows true dual-port timing waveforms for the write operation at port A and the read operation at port B, with the Read-During-Write behavior set to **new data**. Registering the RAM outputs delay the `q` outputs by one clock cycle.

Figure 2–13. True Dual-Port Timing Waveform



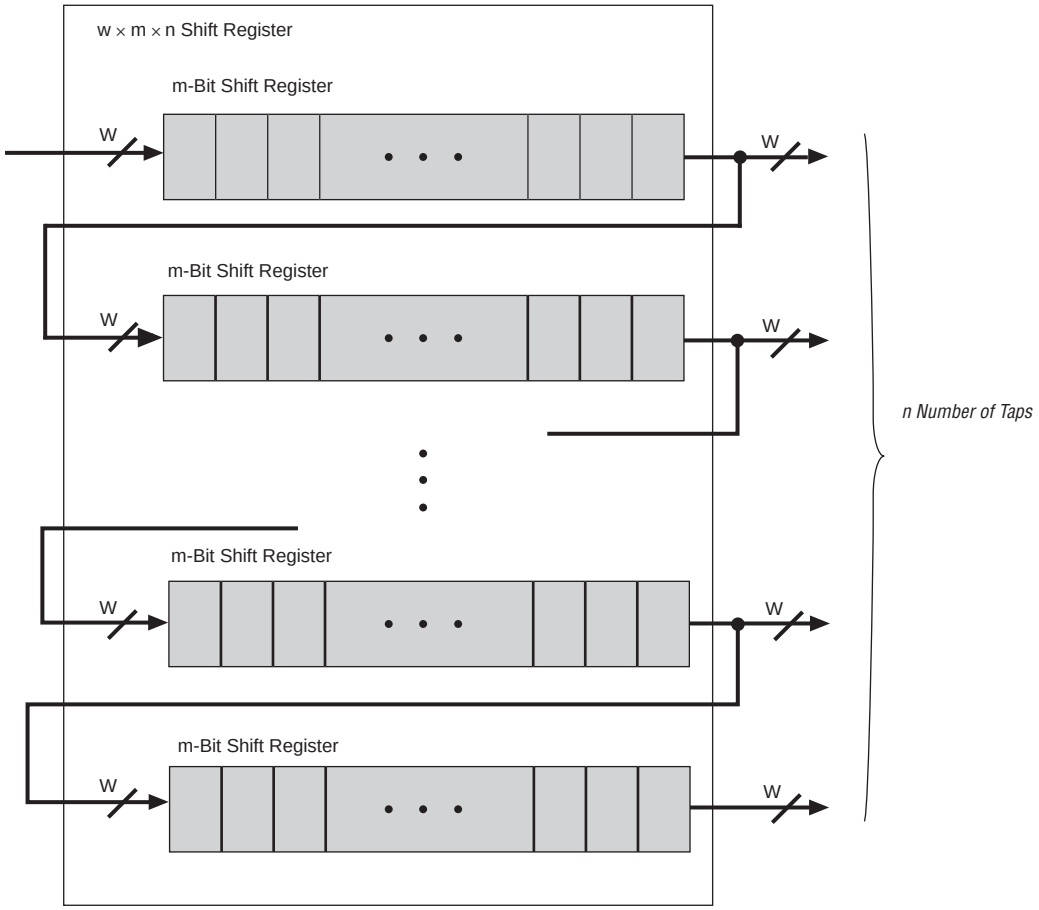
Shift-Register Mode

All Stratix V memory blocks support shift register mode. Embedded memory block configurations can implement shift registers for DSP applications, such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto- and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flipflops that exhaust many logic cells for large shift registers. Alternatively, you can use embedded memory as a shift-register block to save logic cell and routing resources.

The input data width (w), the length of the taps (m), and the number of taps (n) determine the size of a shift register (w x m x n). You can cascade memory blocks to implement larger shift registers.

Figure 2–14 shows the embedded memory block in shift-register mode.

Figure 2–14. Shift-Register Memory Configuration



ROM Mode

All Stratix V embedded memory blocks support ROM mode. A **.mif** initializes the ROM contents of these blocks. The address lines of the ROM are registered on M20K blocks; however, they can be unregistered on MLABs. The outputs can be registered or unregistered. Output registers can be asynchronously cleared. The ROM read operation is identical to the read operation in the single-port RAM configuration.

FIFO Mode

All Stratix V embedded memory blocks support FIFO mode. MLABs are ideal for designs with many small, shallow FIFO buffers. You can use the FIFO MegaWizard Plug-In Manager to implement FIFO buffers in your design. The FIFO MegaWizard Plug-In Manager supports single- and dual-clock (asynchronous) FIFO buffers.

For more information about implementing FIFO buffers, refer to the *SCFIFO and DCFIFO Megafunctions User Guide*.

MLABs do not support mixed-width FIFO mode.

Clocking Modes

Stratix V embedded memory blocks support the following clocking modes:

- "Independent Clock Mode" on page 2–17
- "Input/Output Clock Mode" on page 2–17
- "Read/Write Clock Mode" on page 2–17
- "Single Clock Mode" on page 2–18

Violating the setup or hold time on the memory block address registers could corrupt memory contents. This applies to both read and write operations.

Table 2–9 lists the internal memory clock modes.

Table 2–9. Internal Memory Clock Modes

Clocking Mode	True Dual-Port Mode	Simple Dual-Port Mode	Single-Port Mode	ROM Mode	FIFO Mode
Independent	Y	—	—	Y	—
Input/output	Y	Y	Y	Y	—
Read/write	—	Y	—	—	Y
Single clock	Y	Y	Y	Y	Y

Independent Clock Mode

Stratix V embedded memory blocks can implement independent clock mode for true dual-port memories. In this mode, a separate clock is available for each port (clock A and clock B). Clock A controls all registers on the port A side; clock B controls all registers on the port B side. Each port also supports independent clock enables for both port A and port B registers, respectively. Asynchronous clears are supported only for output latches and output registers on both ports.

Input/Output Clock Mode

Stratix V embedded memory blocks can implement input/output clock mode for true dual-port and simple dual-port memories. In this mode, an input clock controls all registers related to the data input to the memory block including data, address, byte enables, read enables, and write enables. An output clock controls the data output registers. Asynchronous clears are available on output latches and output registers only.

Read/Write Clock Mode

Stratix V embedded memory blocks can implement read/write clock mode for simple dual-port memories. In this mode, a write clock controls the data-input, write-address, and write-enable registers. Similarly, a read clock controls the data-output, read-address, and read-enable registers. The memory blocks support independent clock enables for both the read and write clocks. Asynchronous clears are available on data output latches and registers only.

When using read/write clock mode, if you perform a simultaneous read/write to the same address location, the output read data is unknown. If you require the output data to be a known value, use either single-clock mode or input/output clock mode and choose the appropriate read-during-write behavior in the MegaWizard Plug-In Manager.

Single Clock Mode

Stratix V embedded memory blocks can implement single-clock mode for true dual-port, simple dual-port, and single-port memories. In this mode, a single clock, together with a clock enable, controls all registers of the memory block. Asynchronous clears are available on output latches and output registers only.

Design Considerations

This section describes guidelines for designing with embedded memory blocks.

Selecting Embedded Memory Blocks

The Quartus II software automatically partitions user-defined memory into embedded memory blocks by taking into account both speed and size constraints placed on your design. For example, the Quartus II software may spread memory out across multiple memory blocks when resources are available to increase the performance of the design. You can manually assign memory to a specific block size using the RAM MegaWizard Plug-In Manager.

MLABs can implement single-port SRAM through emulation with the Quartus II software. Emulation results in minimal additional logic resources used. Because of the dual-purpose architecture of the MLAB, it only has data input registers and output registers in the block. MLABs gain read address registers from ALMs, while the write address and read data registers are internal to MLAB.

For more information about register packing, refer to the *Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices* chapter.

Conflict Resolution

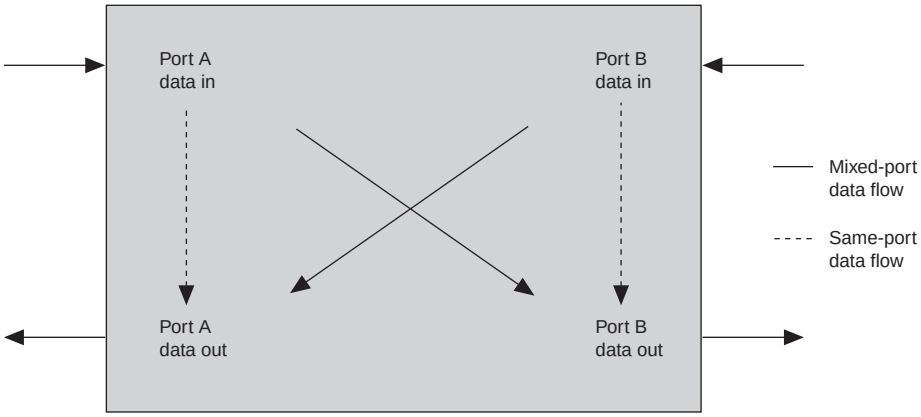
When using memory blocks in true dual-port mode, you can perform two write operations to the same memory location (address). Because there is no conflict resolution circuitry in the memory blocks, you must implement conflict resolution logic external to the memory block to avoid unknown data being written to the address.

Read-During-Write Behavior

You can customize the read-during-write behavior of the Stratix V embedded memory blocks to suit your design requirements. There are two types of read-during-write operations—same port and mixed port.

Figure 2–15 shows the difference between the two types.

Figure 2–15. Read-During-Write Data Flow for Stratix V Devices

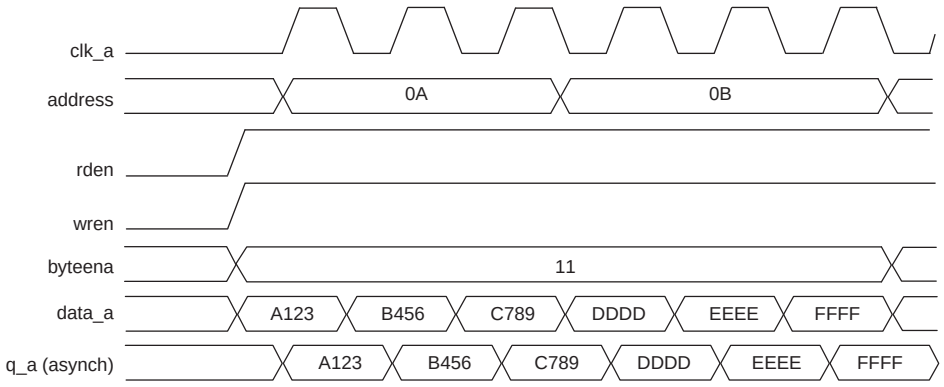


Same-Port Read-During-Write Mode

This mode applies to either a single-port RAM or the same port of a true dual-port RAM. In same-port read-during-write mode, two output choices are available—new data mode (or flow-through) or don't care mode. If the MLAB is selected in same-port read-during-write mode, only the don't care mode is available. In new data mode, the new data is available on the rising edge of the same clock cycle on which it was written. In don't care mode, the RAM outputs "don't care" values for a read-during-write operation.

Figure 2–16 shows sample functional waveforms of same-port read-during-write behavior in new data mode.

Figure 2–16. Same-Port Read-During-Write—New Data Mode



Mixed-Port Read-During-Write Mode

This mode applies to a RAM in simple or true dual-port mode that has one port reading from and the other port writing to the same address location with the same clock.

In this mode, you also have three output choices—"new data", "old data", or "don't care". In old data mode, a read-during-write operation to different ports causes the RAM outputs to reflect the "old data" value at that address location. In don't care mode, the same operation results in a "don't care" or "unknown" value on the RAM outputs.

The RAM MegaWizard Plug-In Manager controls the read-during-write behavior. For more information, refer to the *Internal Memory (RAM and ROM) Megafunction User Guide*.

Figure 2–17 shows a sample functional waveform of mixed-port read-during-write behavior for old data mode.

Figure 2–17. Mixed-Port Read-During-Write—Old Data Mode

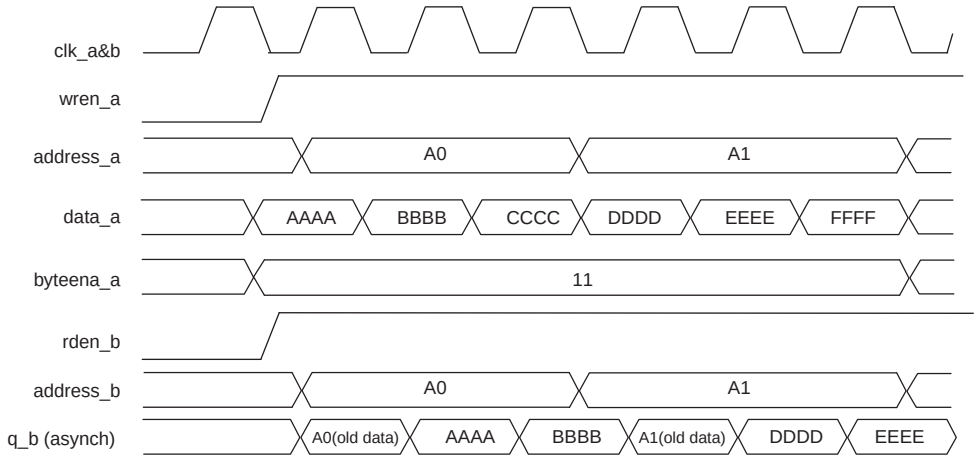
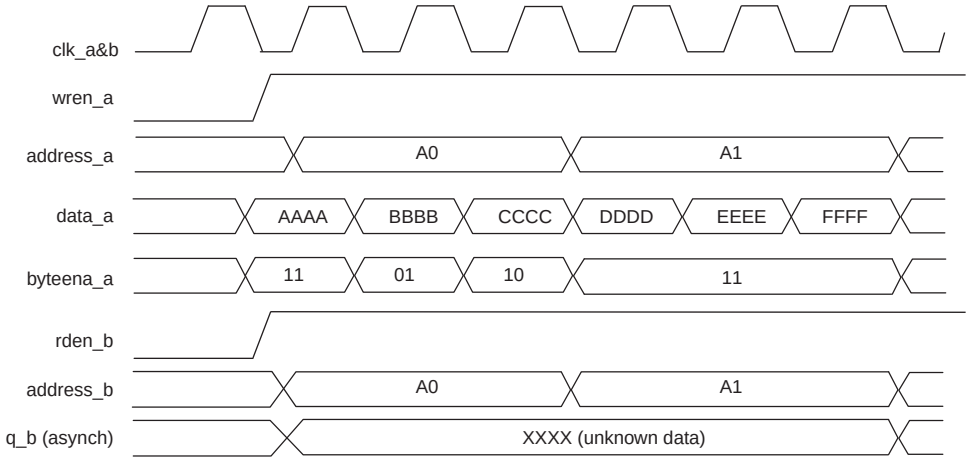


Figure 2–18 shows a sample functional waveform of mixed-port read-during-write behavior for don't care mode.

Figure 2–18. Mixed-Port Read-During-Write—Don't Care Mode



Mixed-port read-during-write is not supported if you use two different clocks in a dual-port RAM. The output value is unknown during a dual-clock mixed-port read-during-write operation.

Power-Up Conditions and Memory Initialization

The M20K memory block outputs power up to zero (cleared), regardless of whether the output registers are used or bypassed. MLABs power up to zero if the output registers are used and power up reading the memory contents if the output registers are not used. You must consider this when designing logic that might evaluate the initial power-up values of the MLAB memory block. For Stratix V devices, the Quartus II software initializes the RAM cells to zero unless there is a **.mif** specified.

All memory blocks support initialization with a **.mif**. You can create **.mif** files in the Quartus II software and specify their use with the RAM MegaWizard Plug-In Manager when instantiating a memory in your design. Even if a memory is pre-initialized (for example, using a **.mif**), it still powers up with its outputs cleared.

For more information about **.mif** files, refer to the *Internal Memory (RAM and ROM) Megafunction User Guide* and the *Quartus II Handbook*.

Power Management

Stratix V memory block clock-enables allow you to control the clocking of each memory block to reduce AC power consumption. Use the read-enable signal to ensure that read operations only occur when you require read operations. If your design does not require read-during-write, you can reduce your power consumption by deasserting the read-enable signal during write operations, or any period when no memory operations occur.

The Quartus II software automatically places any unused memory blocks in low-power mode to reduce static power.

Document Revision History

Table 2–10 lists the revision history for this chapter.

Table 2–10. Document Revision History

Date	Version	Changes
June 2012	1.4	Updated Table 2–1 and Table 2–2.
November 2011	1.3	■ Updated Table 2–1 and Table 2–2. ■ Updated “Mixed-Port Read-During-Write Mode” section.
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 2–1, Table 2–2, and Table 2–5. ■ Updated Figure 2–1 and Figure 2–8. ■ Updated “Read-During-Write Behavior” section. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

3. Variable Precision DSP Blocks in Stratix V Devices

SV51004-1.4

This chapter describes how the variable precision digital signal processing (DSP) blocks in Stratix® V devices are optimized to support higher-bit precision in high-performance DSP applications, such as radar systems that must support higher resolution and multi-antenna architectures, wireless-base station channel cards for multiple-input multiple-output (MIMO) processing, medical and test applications for very high-precision filtering, and fast Fourier transforms (FFTs) functions.

You can configure a variable precision DSP block to implement one of several operational modes to suit your application. The built-in pre-adder, coefficient bank, multipliers, and adder/subtractor minimize the amount of external logic to implement these functions, resulting in efficient resource usage, reduced power consumption, improved performance, and data throughput for DSP applications.

A Stratix V variable precision DSP block maintains some backward compatibility features, so it can efficiently support existing 18-bit DSP applications, such as high-definition video processing, digital-up and down conversion, and multi-rate filtering.

This chapter contains the following sections:

- "Variable Precision DSP Block Overview" on page 3–1
- "Operational Modes Overview" on page 3–3
- "Variable Precision DSP Block Resource Descriptions" on page 3–4
- "Operational Mode Descriptions" on page 3–9
- "Software Support" on page 3–23

Variable Precision DSP Block Overview

Each Stratix V variable precision DSP block spans one logic array block (LAB) row height.

The following are the architectural highlights of the Stratix V variable precision DSP block:

- High-performance, power-optimized, and fully registered multiplication operations
- Natively supported 9-bit, 18-bit, 27-bit, and 36-bit word lengths
- Efficiently supported 18 x 25 complex multiplications for FFTs
- Efficiently supported floating-point arithmetic formats
- Built-in addition, subtraction, and 64-bit accumulation units to combine multiplication results efficiently

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

- Cascading 18-bit and 27-bit input bus to form tap-delay line for filtering applications
- Cascading 64-bit output bus to propagate output results from one block to the next block without external logic support
- Hard pre-adder supported in 18-bit and 27-bit mode for symmetric filters
- Supports 18-bit and 27-bit with internal coefficient register bank for filter implementation
- Efficiently supported 18-bit or 27-bit systolic finite impulse response (FIR) filters with distributed output adder

Table 3–1 lists the number of multipliers in Stratix V devices.

Table 3–1. Number of Multipliers in Stratix V Devices

Family	Device	Variable Precision DSP Blocks	Independent Input and Output Multiplication Operators					18 × 18 Multiplier Adder Mode	18 × 18 Multiplier Summed with 36-Bit input
			9 × 9 Multipliers	16 × 16 Multipliers	18 × 18 Multipliers with 32-Bit Resolution	27 × 27 Multipliers	36 × 18 Multipliers		
Stratix V GX	5SGXA3	256	768	512	512	256	256	512	256
	5SGXA4	256	768	512	512	256	256	512	256
	5SGXA5	256	768	512	512	256	256	512	256
	5SGXA7	256	768	512	512	256	256	512	256
	5SGXA9	352	1,056	704	704	352	352	704	352
	5SGXAB	352	1,056	704	704	352	352	704	352
	5SGXB5	399	1,197	798	798	399	399	798	399
	5SGXB6	399	1,197	798	798	399	399	798	399
	5SGXB9	352	1,056	704	704	352	352	704	352
	5SGXBB	352	1,056	704	704	352	352	704	352
Stratix V GT	5SGTC5	256	768	512	512	256	256	512	256
	5SGTC7	256	768	512	512	256	256	512	256
Stratix V GS	5SGSD3	600	1,800	1,200	1,200	600	600	1,200	600
	5SGSD4	1,044	3,132	2,088	2,088	1,044	1,044	2,088	1,044
	5SGSD5	1,590	4,770	3,180	3,180	1,590	1,590	3,180	1,590
	5SGSD6	1,775	5,325	3,550	3,550	1,775	1,775	3,550	1,775
	5SGSD8	1,963	5,889	3,926	3,926	1,963	1,963	3,926	1,963
Stratix V E	5SEE9	352	1,056	704	704	352	352	704	352
	5SEEB	352	1,056	704	704	352	352	704	352

Operational Modes Overview

Table 3–2 lists the summary of the operational modes that are supported by Stratix V variable precision DSP blocks.

Table 3–2. Variable Precision DSP Blocks Operational Modes for Stratix V Devices

Variable Precision DSP Block Resources	Operational Mode	Supported Instance	Pre-adder Support	Coefficient Support	Input Cascade support	Chainout Support
1 variable precision DSP block	Independent 9 x 9 Multiplication	3	No	No	No	No
	Independent 16 x 16 Multiplication	2	Yes	Yes	Yes	No
	Independent 18 x 18 Partial Multiplication (32-Bit)	2	Yes	Yes	Yes	No
	Independent 18 x 18 Multiplication	1	Yes	Yes	Yes	No
	Independent 27 x 27 Multiplication	1	Yes	Yes	Yes	Yes
	Independent 36 x 18 Multiplication	1	No	Yes	No	Yes
	Two 18 x 18 Multiplier Adder	1	Yes	Yes	Yes	Yes
	Two 16 x 16 Multiplier Adder	1	Yes	Yes	Yes	Yes
	Sum of 2 Square	1	Yes (1)	No	No	Yes
	18 x 18 Multiplication Summed with 36-Bit Input	1	No	No	No	Yes
2 variable precision DSP blocks	Independent 18 x 18 Multiplication	3	No	No	No	No
	Independent 36 x 36 Multiplication	1	No	No	No	No
	Complex 18 x 18 Multiplication	1	Yes	Yes	Yes	Yes
	Four 18 x 18 Multiplier Adder	1	Yes	Yes	Yes	No
	Two 27 x 27 Multiplier Adder	1	Yes	Yes	Yes	No
	Two 18 x 36 Multiplier Adder	1	No	Yes	No	No
3 variable precision DSP blocks	Complex 18 x 25 Multiplication	1	Yes (1)	No	No	No
4 variable precision DSP blocks	Complex 27 x 27 Multiplication	1	Yes	Yes	Yes	No

Note to Table 3–2:

(1) The pre-adder feature for this mode is automatically enabled.

The Quartus® II software includes megafunctions that you can use to control the operation mode of the multipliers. After making the appropriate parameter settings with the MegaWizard™ Plug-In Manager, the Quartus II software automatically configures the variable precision DSP block.

Variable Precision DSP Block Resource Descriptions

The Stratix V variable precision DSP block consists of the following elements:

- Input register bank
- Pre-adder and coefficient select
- Multipliers
- Chainout adder and accumulator
- Systolic registers
- Output register bank

Figure 3–1 shows a detailed overall architecture of the 18 x 18 Stratix V variable precision DSP block.

Figure 3–1. 18 x 18 Variable Precision DSP Block Architecture for Stratix V Devices

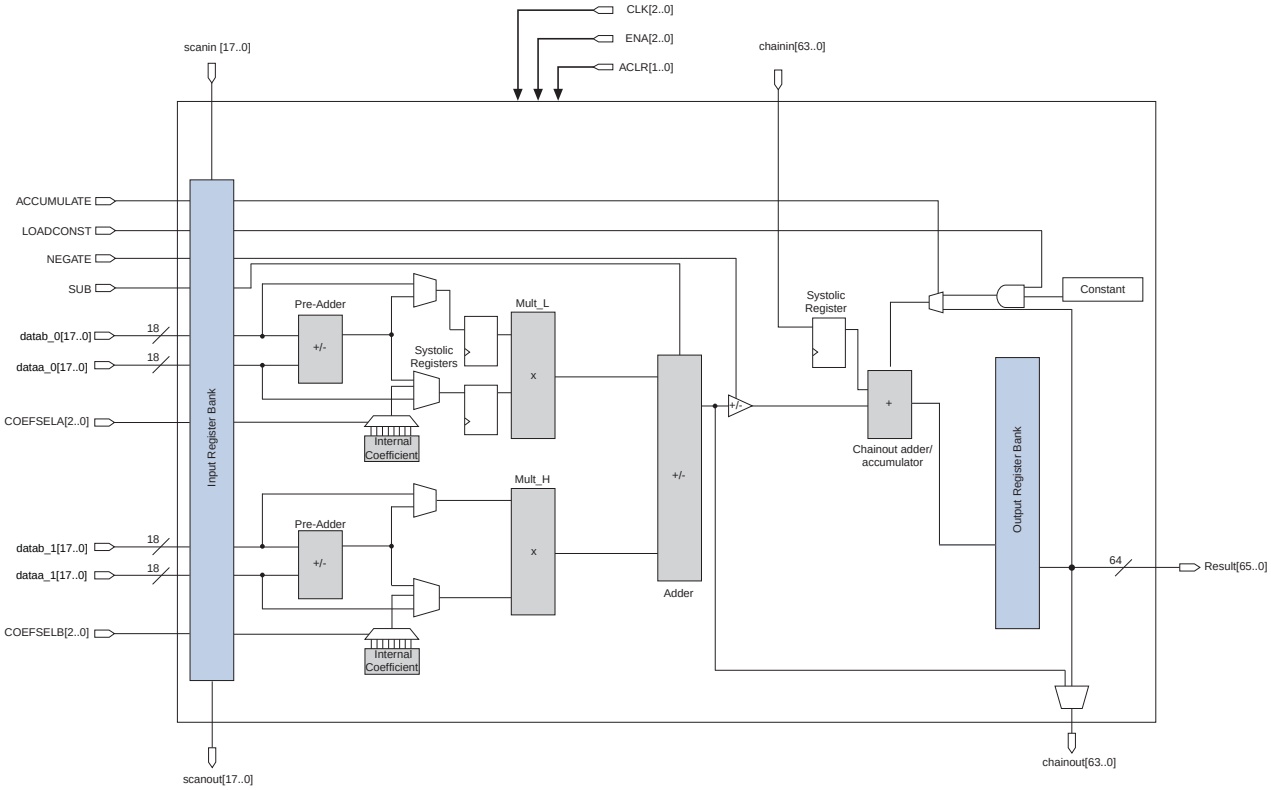
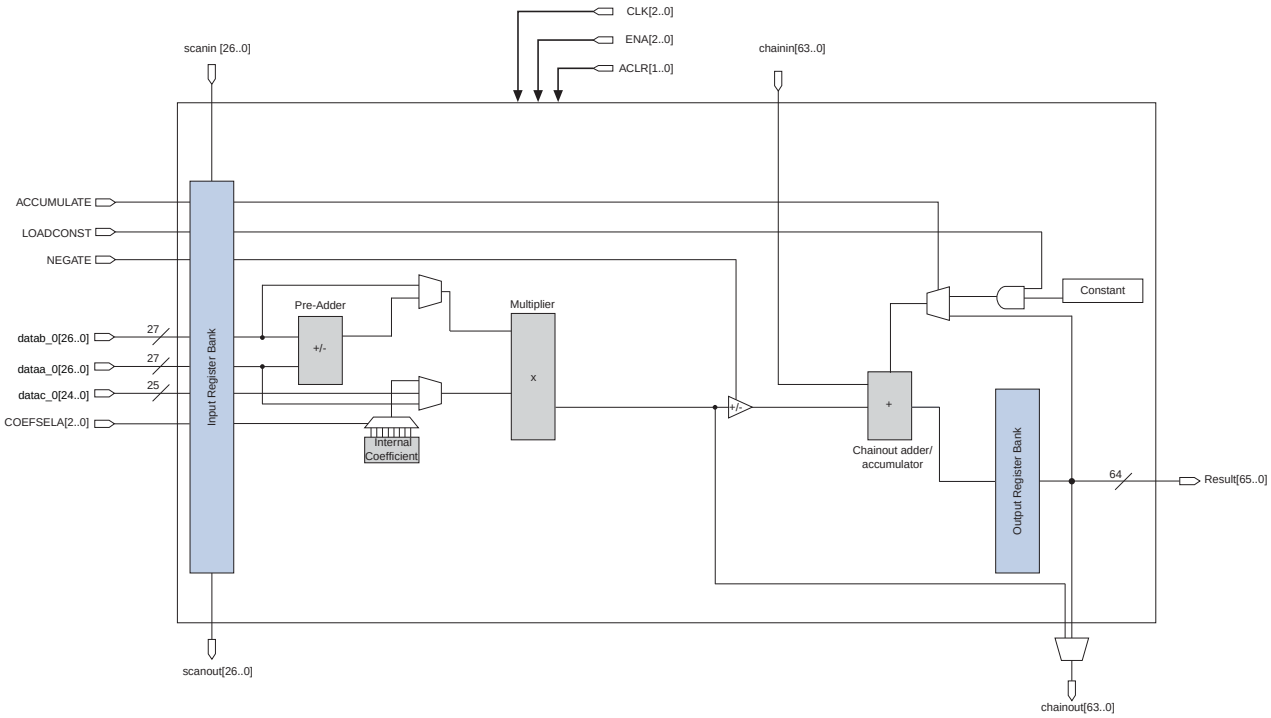


Figure 3–2 shows a detailed overall architecture of the 27 x 27 Stratix V variable precision DSP block.

Figure 3–2. 27 x27 Variable Precision DSP Block Architecture for Stratix V Devices



Input Registers Bank

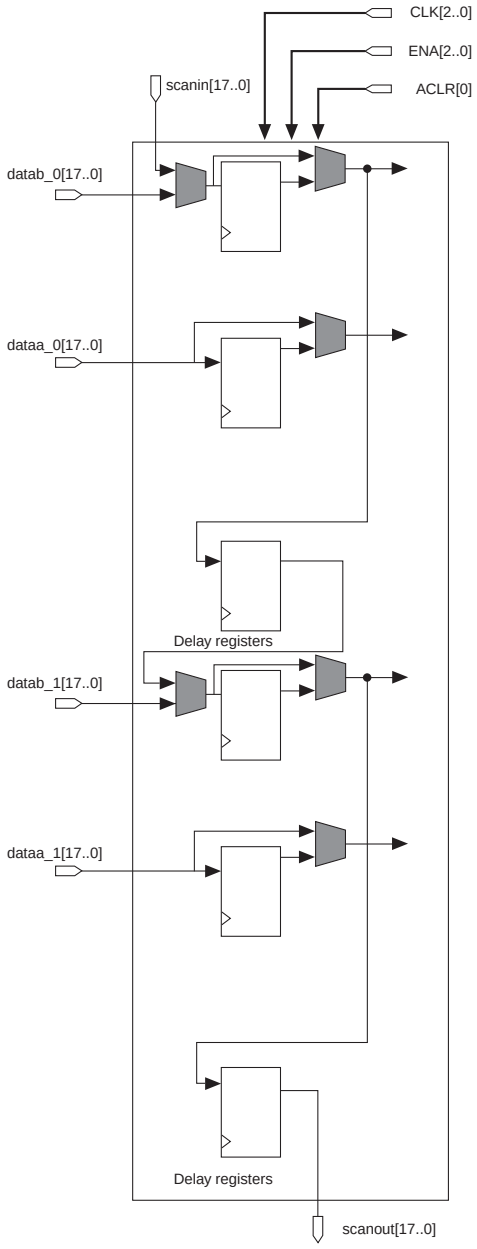
The positive edge of the clock signal triggers all variable precision DSP block registers and clears them after power up. Each multiplier operand can feed an input register or a multiplier directly, bypassing the input registers. The following variable precision DSP block signals control the input registers within the variable precision DSP block:

- `CLK[2..0]`
- `ENA[2..0]`
- `ACLR[0]`

Besides the registers for the data and dynamic control signals, there are also two sets of delay registers in the input register bank. The delay registers are used to balance the latency requirements when both the input cascade and chainout features are used. This is only supported in 18 x 18 mode.

One feature of the input register bank is to support a tap delay line; therefore, you can drive the top leg of the multiplier input (B) from general routing or from the cascade chain, as shown in Figure 3–3 and Figure 3–4. The Stratix V variable precision DSP block supports 18-bit and 27-bit input cascading.

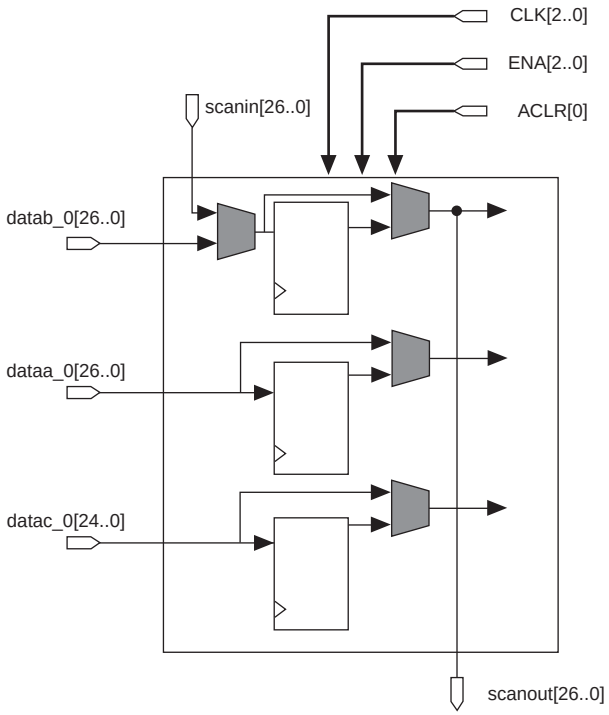
Figure 3–3. Input Register of a Variable Precision DSP Block in 18 x 18 Mode for Stratix V Devices



Note to Figure 3–3:

(1) Figure 3–3 shows only the data registers. Registers for the control signals are not shown.

Figure 3–4. Input Register of a Variable Precision DSP Block in 27 x 27 Mode for Stratix V Devices



Note to Figure 3–4:

(1) Figure 3–4 shows only the data registers. Registers for the control signals are not shown.

Pre-Adder and Coefficient Select

The pre-adder supports both addition and subtraction, which you must choose during compilation time. There are two 18-bit pre-adders in each variable precision DSP block. You can configure these two pre-adders as two independent 18-bit adders for 18-bit applications, or a 26-bit adder for 27-bit applications.

The Stratix V variable precision DSP block has the flexibility of selecting the multiplicand from either the dynamic input or internal coefficient. The internal coefficient can support up to eight constant coefficients for the multiplicands in 18-bit and 27-bit applications. When you enable the internal coefficient feature, the `COEFSELA/COEFSELB` are used to control the dynamic selection of the coefficient multiplexer.

In 18-bit applications, you must enable the coefficient feature when the pre-adder feature is enabled. In 27-bit applications, you can use the coefficient feature and pre-adder feature independently. In 27-bit applications with the pre-adder feature enabled, the input data width is restricted to 22 bits if the multiplicand input comes from dynamic input due to input limitations. If the multiplicand input comes from internal coefficient, the data width of the input is 27 bits.

When you enable the pre-adder feature, all input data must have the same clock setting.

Multipliers

There are two multipliers (Mult_H and Mult_L) per variable precision DSP block. You can configure these two multipliers to work as a 27 x 27 multiplier, two 18 x 18 multipliers, or three 9 x 9 multipliers, depending on the operational mode. A single variable precision DSP block can perform many multiplications in parallel, depending on the data width of the multiplier. For more information, refer to "Operational Mode Descriptions" on page 3–9.

Chainout Adder and Accumulator

The accumulator feature is not available in multi-block modes.

The Stratix V variable precision DSP blocks contain a 64-bit adder and a 64-bit accumulator. You can use the 64-bit adder as full adder.

Table 3–3 lists the functions supported by the accumulator in Stratix V devices.

Table 3–3. Functions Supported by Accumulator in Stratix V Devices

Function	Description
Zeroing	Disables the accumulator.
Preload	Loads an initial value to the accumulator. Only 1 bit of the 64-bit preload value can be "1". It can be used as rounding the DSP result to any position of the 64-bit result.
Accumulation	Adds the current result to the previous accumulate result.
Decimation	This function takes the current result, converts it into two's compliment, and adds it to the previous result.

You can dynamically control the function of the accumulator by three control signals—NEGATE, LOADCONST, and ACCUMULATE. Table 3–4 lists how these dynamic signals control the accumulator functions.

Table 3–4. Dynamic Control Signals for 64-Bit Accumulator for Stratix V Devices

Function	NEGATE	LOADCONST	ACCUMULATE
Zero	0	0	0
Accumulate	0	0	1
Decimate	1	0	1
Preload	0	1	0

Systolic Register

There are two systolic registers per variable precision DSP block. The first systolic register has two 18-bit registers that are used to register the Mult_L's two 18-bit inputs. You must clock these registers with the same clock source as the output register bank. The second systolic register is used to delay the chainout output to the next variable precision DSP block. If the variable precision DSP block is not configured in systolic FIR mode, both systolic registers are bypassed.

Output Register Bank

The positive edge of the clock signal triggers the 64-bit bypassable output register bank and is cleared after power up. The following variable precision DSP block signals control the output register per variable precision DSP block:

- `CLK[2..0]`
- `ENA[2..0]`
- `ACLR[1]`

Operational Mode Descriptions

This section describes how you can configure a Stratix V variable precision DSP block to efficiently support the following operational modes:

- "Independent Multiplier Modes" on page 3–9
- "Independent Complex Multiplier Modes" on page 3–14
- "Multiplier Adder Sum Mode" on page 3–17
- "Sum of Square Mode" on page 3–20
- "18 x 18 Multiplication Summed with 36-Bit Input Mode" on page 3–20
- "Systolic FIR Mode" on page 3–21

Independent Multiplier Modes

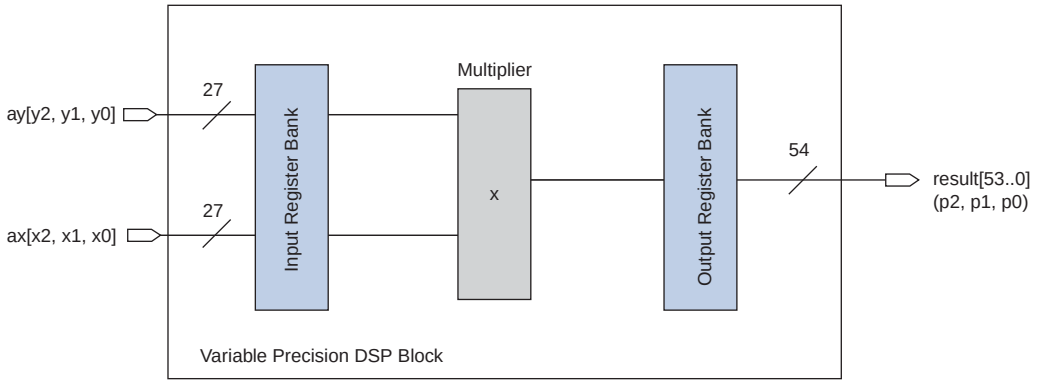
In independent input and output multiplier mode, the variable precision DSP blocks perform individual multiplication operations for general purpose multipliers.

9 x 9, 16 x 16, 18 x18, 27 x 27, and 36 x 18 Multipliers

You can configure each variable precision DSP block multiplier for 9-, 16-, 18-, 27-bit, or 36 x 18 multiplication. A variable precision DSP block can support up to three individual 9 x 9 multipliers, two individual 16 × 16 multipliers, two individual 18 x 18 partial multipliers, one individual 18 x 18 multiplier, one individual 27 x 27 multiplier, or one individual 36 x 18 multiplier. For some operational modes, the unused inputs require zero padding.

Figure 3–5, Figure 3–6, Figure 3–8 on page 3–12, Figure 3–9 on page 3–12, and Figure 3–10 on page 3–13 show the variable precision DSP block in independent multiplier operation mode. Figure 3–7 on page 3–11 shows that two variable precision DSP blocks can support three individual 18 x 18 multipliers.

Figure 3–5. Three 9 x 9 Independent Multiplier Mode for Stratix V Devices *(1)*



Note to Figure 3–5:

(1) Three pairs of data are packed into the ax and ay ports; result contains three 18-bit products.

Figure 3–6. One 18 x 18 Independent Multiplier Mode with One Variable Precision DSP Block for Stratix V Devices

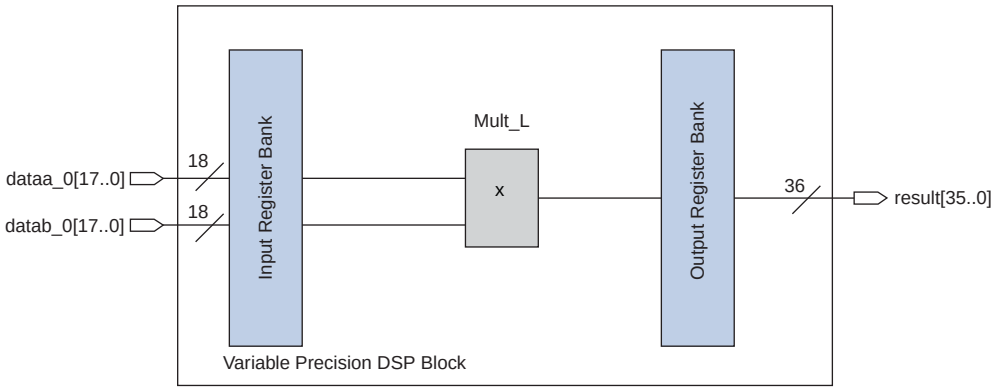


Figure 3–7. Three 18 x 18 Independent Multiplier Mode with Two Variable Precision DSP Blocks for Stratix V Devices

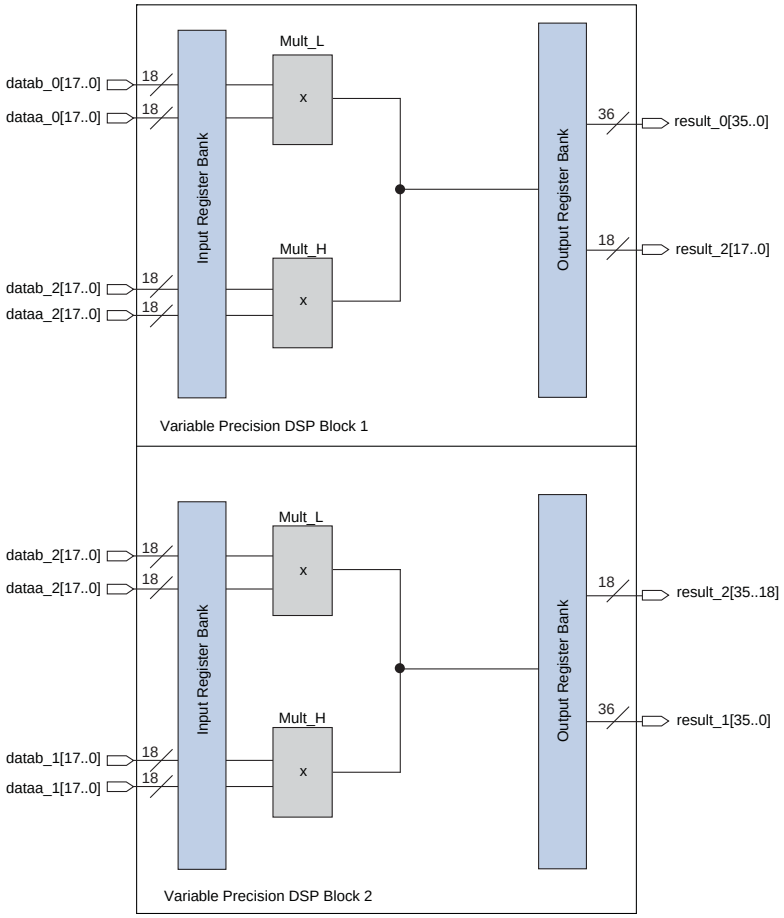
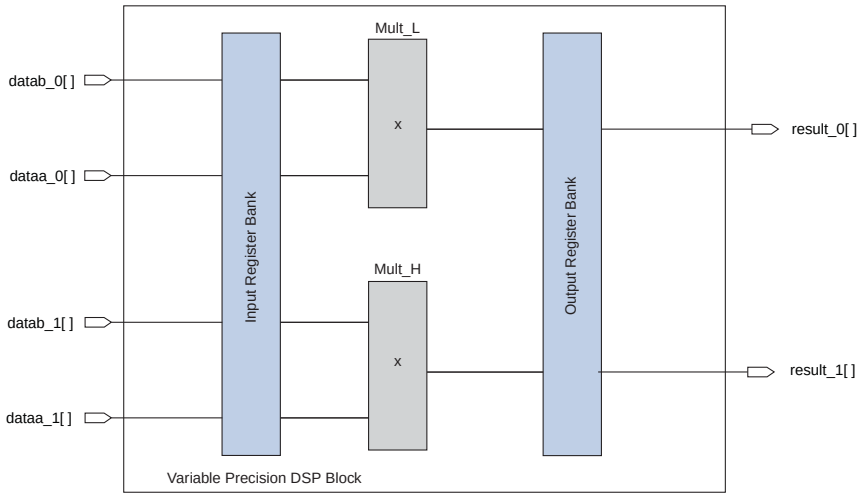


Figure 3–8. Two 16 x 16 Independent Multiplier Mode or Two 18 x 18 Independent Partial Multiplier Mode for Stratix V Devices (1), (2)

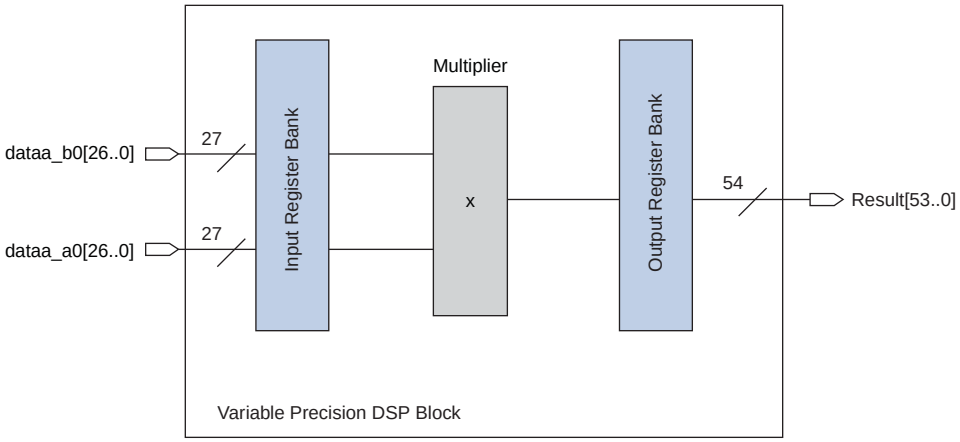


Notes to Figure 3–8:

(1) The inputs for 16-bit independent multiplier mode are `data[15..0]`. The unused input bits require padding with zero.

(2) For 18-bit independent multiplier mode, only 32-LSB output for both multipliers are routed to the output register.

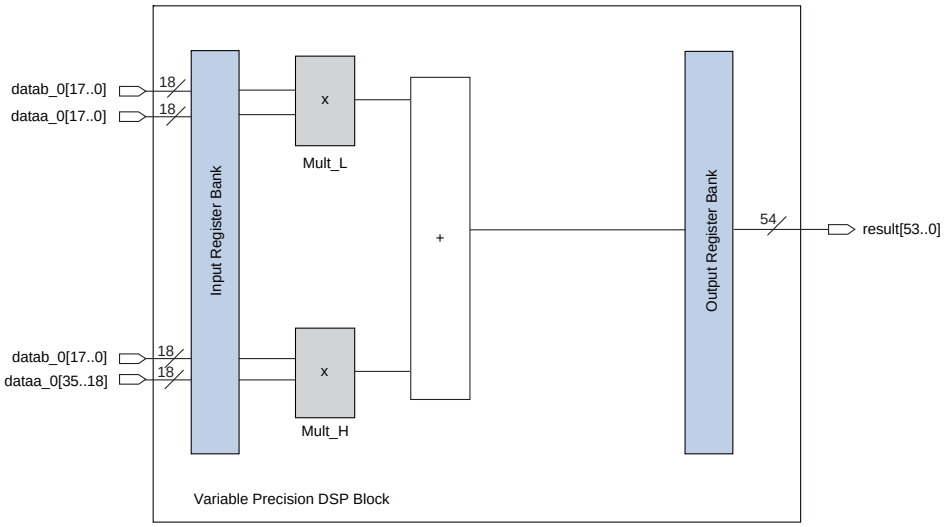
Figure 3–9. One 27 x 27 Independent Multiplier Mode for Stratix V Devices (1)



Note to Figure 3–9:

(1) The result can be up to 64-bits when combined with a chainout adder/accumulator.

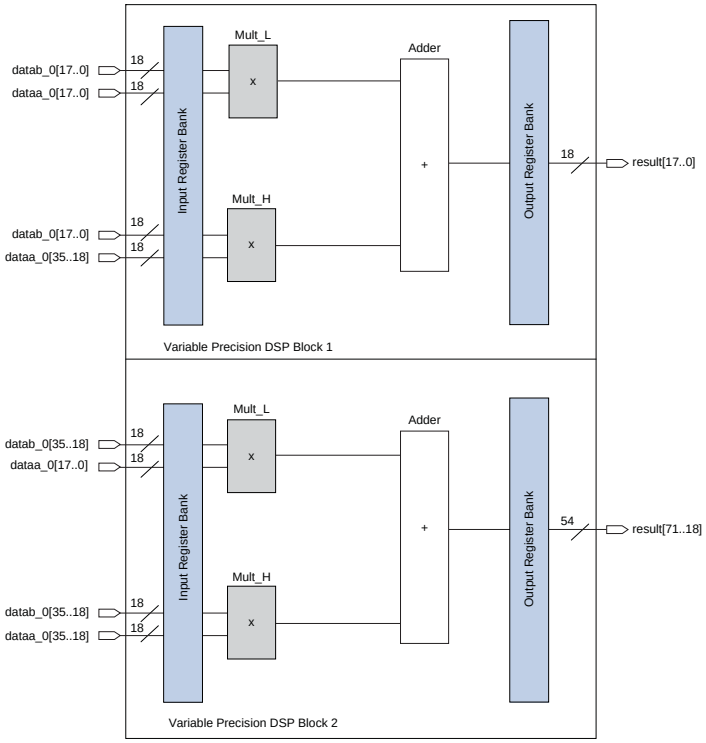
Figure 3–10. One 36 × 18 Independent Multiplier Mode for Stratix V Devices



36-Bit Multiplier

You can efficiently construct an individual 36-bit multiplier with two adjacent Stratix V variable precision DSP blocks. The 36 x 36 multiplication consists of four 18 x 18 multipliers, as shown in Figure 3–11. The 36-bit multiplier is useful for applications requiring more than 18-bit precision; for example, for the mantissa multiplication portion of very high precision fixed-point arithmetic applications.

Figure 3–11. 36-Bit Independent Multiplier Mode with Two Variable Precision DSP Blocks for Stratix V Devices



Independent Complex Multiplier Modes

The Stratix V variable precision DSP block provides the means for a complex multiplication. Equation 3–1 shows a complex multiplication that you can write.

Equation 3–1. Complex Multiplication Equation

$$(a + jb) \times (c + jd) = [(a \times c) - (b \times d)] + j[(a \times d) + (b \times c)]$$

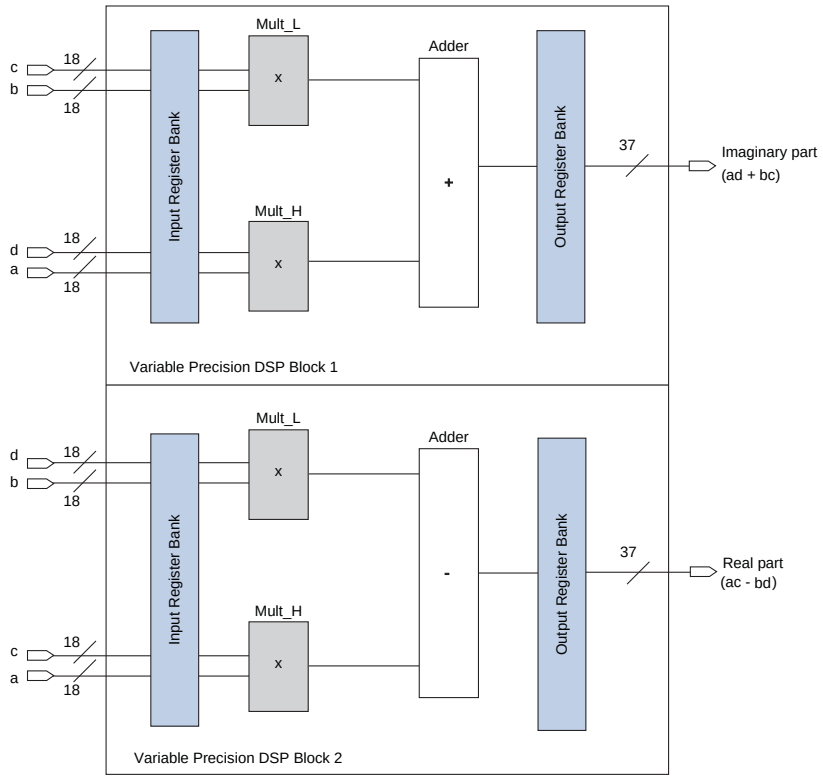
The Stratix V variable precision DSP block can support an 18 x 18 complex multiplier, 18 x 25 complex multiplier, or a 27 x 27 complex multiplier.

18 x 18 Complex Multiplier

For 18 x 18 complex multiplications, you require two variable precision DSP blocks to perform this multiplication mode. You can implement the imaginary part [(a × d) + (b × c)] in the first variable precision DSP block, and you can implement the real part [(a × c) – (b × d)] in the second variable precision DSP block.

Figure 3–12 shows an 18-bit complex multiplication.

Figure 3–12. 18 x 18 Complex Multiplier with Two Variable Precision DSP Blocks for Stratix V Devices



18 x 25 Complex Multiplier

Stratix V devices support an individual 18 x 25 complex multiplication mode. A 27 x 27 multiplier allows you to implement an individual 18 x 25 complex multiplication mode with three variable precision DSP blocks only. The pre-adder feature is automatically enabled for you to implement an individual 18 x 25 complex multiplication mode efficiently.

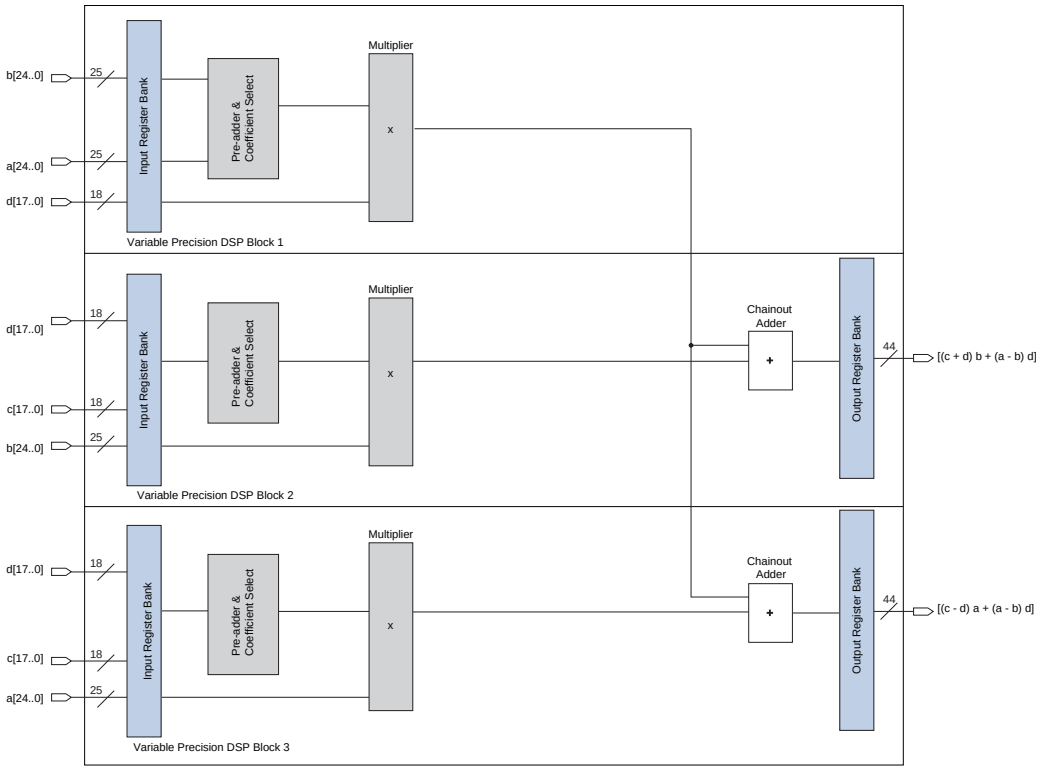
You can implement an 18 x 25 complex multiplication with three variable precision DSP blocks, as shown in Equation 3–2.

Equation 3–2. 18 × 25 Complex Multiplication Equation

$$(a + jb) \times (c + jd) = (c - d) \times a + (a - b) \times d + j\,[(c + d) \times b + (a - b) \times d]$$

Figure 3–13 shows an 18 x 25 complex multiplication with three variable precision DSP blocks.

Figure 3–13. 18 x 25 Complex Multiplier with Three Variable Precision DSP Blocks for Stratix V Devices

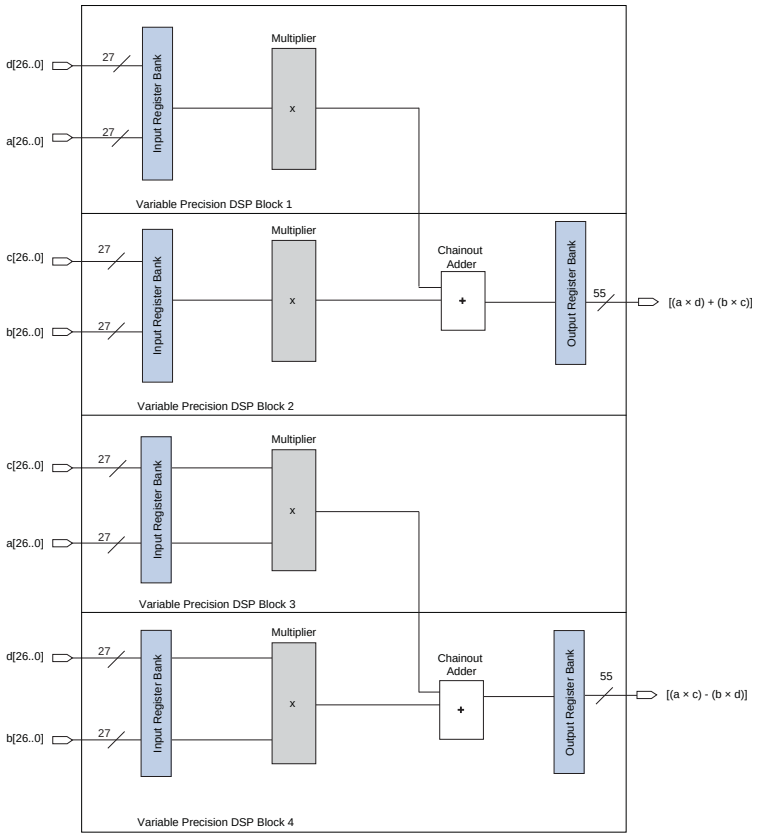


27 x 27 Complex Multiplier

Stratix V devices support an individual 27 x 27 complex multiplication mode. You require four variable precision DSP blocks to implement an individual 27 x 27 complex multiplication mode. You can implement the imaginary part [(a × d) + (b × c)] in the first and second variable precision DSP blocks, and you can implement the real part [(a × c) - (b × d)] in the third and fourth variable precision DSP blocks. You can achieve the difference of two 27 × 27 multiplications by enabling the NEGATE control signal in the fourth variable precision DSP block.

Figure 3–14 shows a 27-bit complex multiplication with four variable precision DSP blocks.

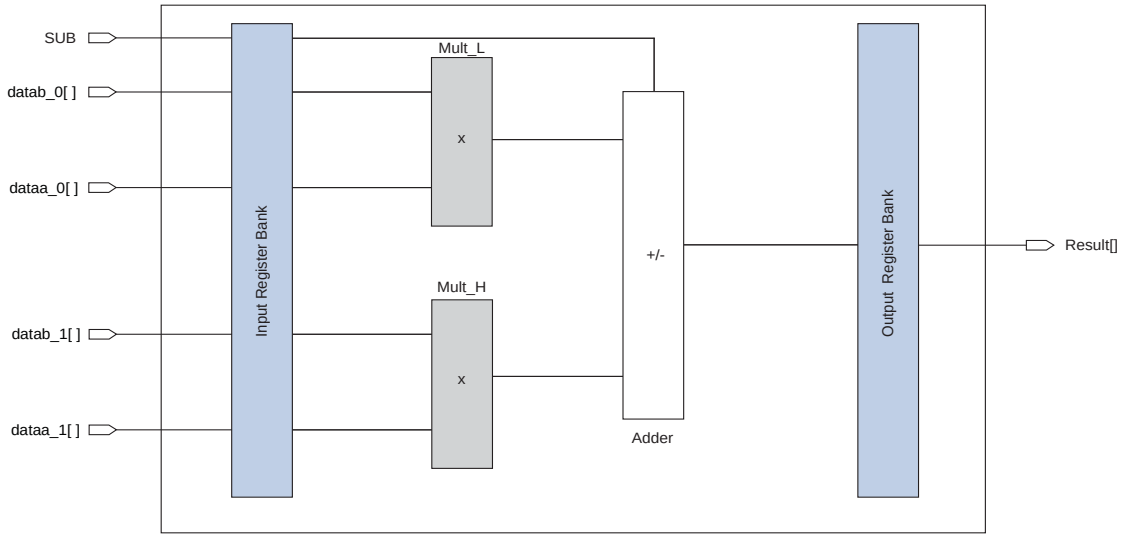
Figure 3–14. 27 × 27 Complex Multiplier with Four Variable Precision Blocks



Multiplier Adder Sum Mode

Stratix V devices support two-multiplier adder sum mode and four-multiplier adder sum mode. For a two-multiplier adder configuration, the Stratix V variable precision DSP blocks can support 16-bit, 18-bit, 27-bit, and 18 × 36 multipliers. You require two variable precision DSP blocks to implement 27-bit and 18 x 36 multiplier adder sum mode. Stratix V devices support one sum of four 18-bit multipliers with two variable precision DSP blocks. Figure 3–15 through Figure 3–18 show the variable precision DSP blocks in the multiplier adder sum mode.

Figure 3–15. One Sum of Two 18 x 18 Multipliers or Two 16 x 16 Multipliers for Stratix V Devices *(1), (2)*



Notes to Figure 3–15:

(1) For 18-bit multiplier adder sum mode, the input data width is 18 bits and the output data width is 37 bits.

(2) For 16-bit multiplier adder sum mode, the input data width is 16 bits and the unused input bit requires padding with zeroes. The output data width is 33 bits.

Figure 3–16. One Sum of Two 27 x 27 Multipliers with Two Variable Precision DSP Blocks for Stratix V Devices

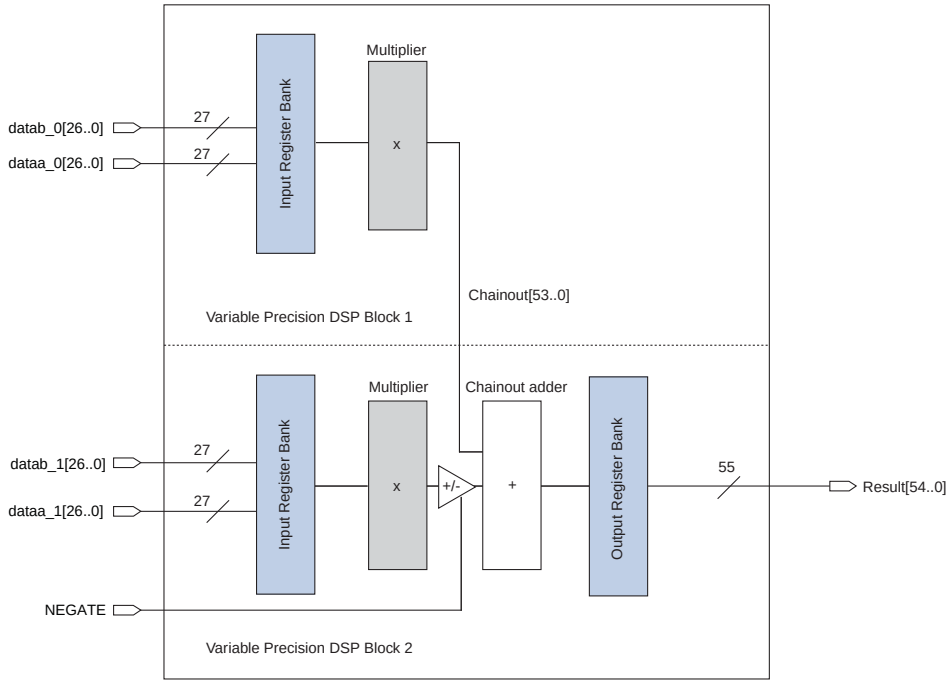


Figure 3–17. One Sum of Two 36 x 18 Multipliers with Two Variable Precision DSP Blocks for Stratix V Devices

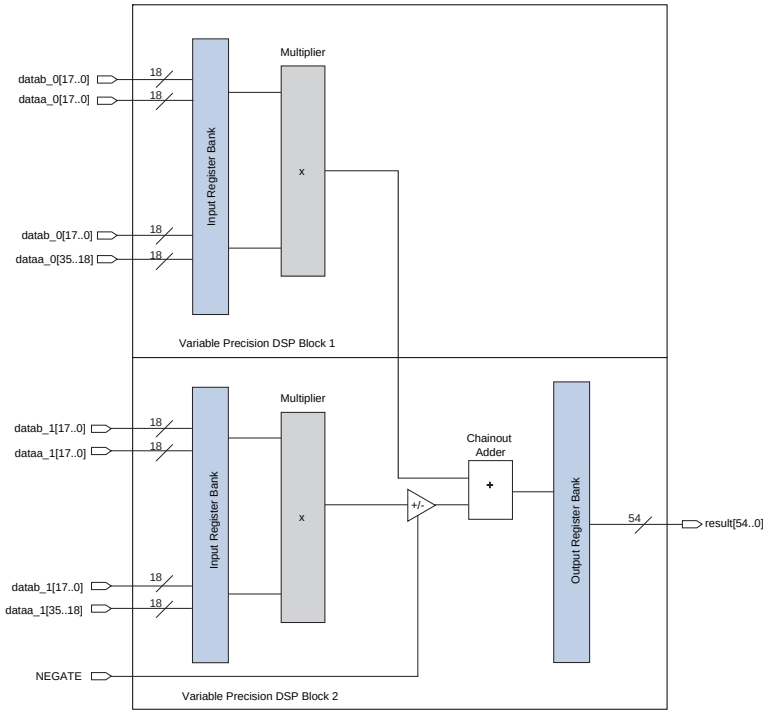
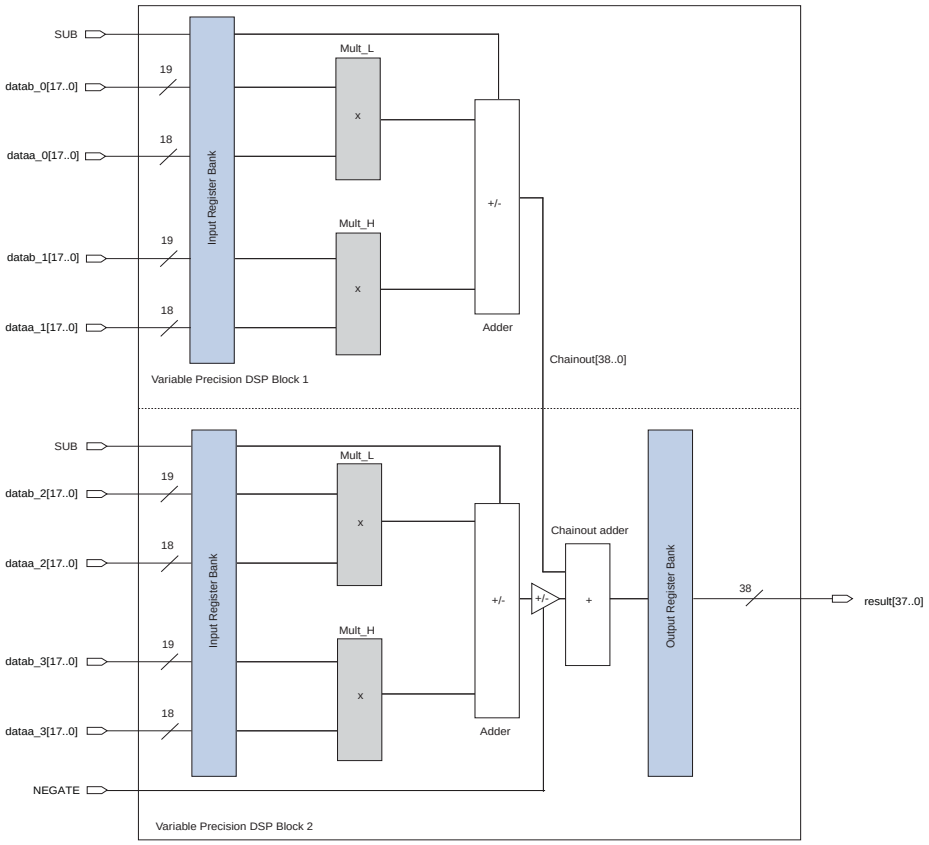


Figure 3–18. One Sum of Four 18 x 18 Multipliers with Two Variable Precision DSP Blocks for Stratix V Devices

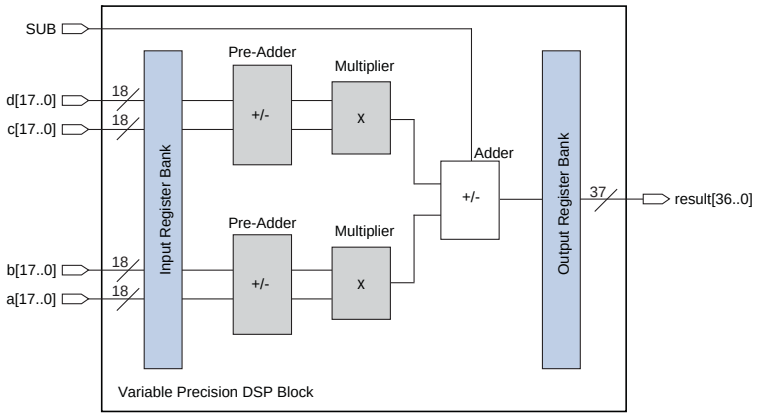


Sum of Square Mode

The Stratix V variable precision DSP block can implement one sum of square mode, $(a \pm b)^2 \times (c \pm d)^2$. You can feed the four 18-bit inputs into the pre-adder block to convert `b` and `d` input as two's complement numbers to perform subtraction, if required. You can feed each 18-bit pre-adder block output into both multiplicand and multiplier inputs of an 18 x 18 multiplier to generate a square result.

Figure 3–19 shows the sum of square mode in a variable precision DSP block.

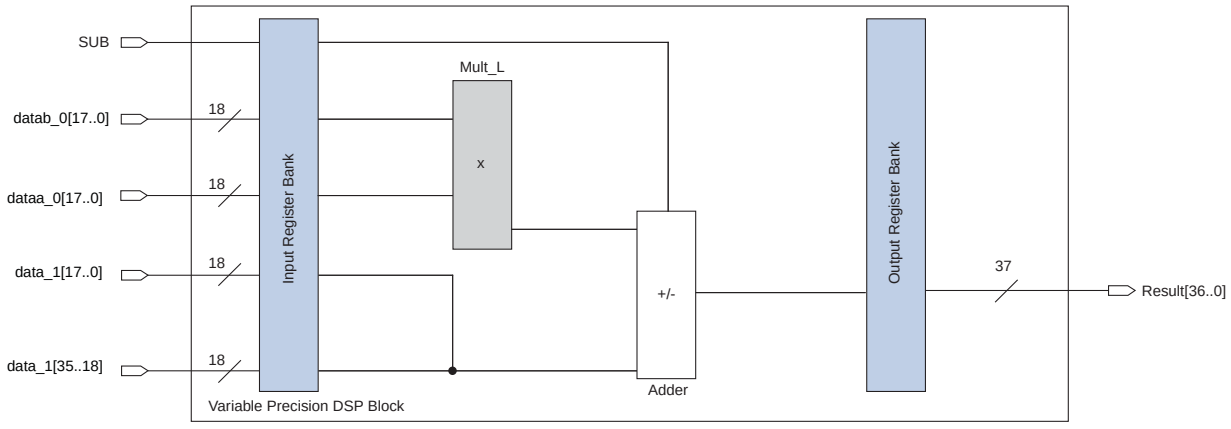
Figure 3–19. One Sum of Square Mode in a Variable Precision DSP Block for Stratix V Devices



18 x 18 Multiplication Summed with 36-Bit Input Mode

Stratix V variable precision DSP blocks support one 18 x 18 multiplication summed to a 36-bit input. You can use Mult_L to provide the input for an 18 x 18 multiplication, whereas Mult_H is bypassed. The `data1[17..0]` and `data1[35..18]` signals are concatenated to produce a 36-bit input. Figure 3–20 shows the 18 x 18 multiplication summed with the 36-bit input mode in a variable precision DSP block.

Figure 3–20. One 18 × 18 Multiplication Summed with 36-Bit Input Mode for Stratix V Devices



Systolic FIR Mode

Stratix V variable precision DSP blocks support 18-bit and 27-bit systolic FIR structures. In systolic FIR mode, the input of the multiplier can come from three different sets of sources:

- two dynamic inputs
- one dynamic input and once coefficient input
- one coefficient input and one pre-adder output

Figure 3–21 shows the 18-bit systolic FIR with two dynamic inputs. In 18-bit systolic FIR mode, the adders are configured as dual 44-bit adders, thereby giving 8 bits of overhead when using an 18-bit operation (36-bit products). This allows a total sum of 256 multiplier products.

Figure 3–21. 18-bit Systolic FIR Mode with Two Dynamic Inputs for Stratix V Devices

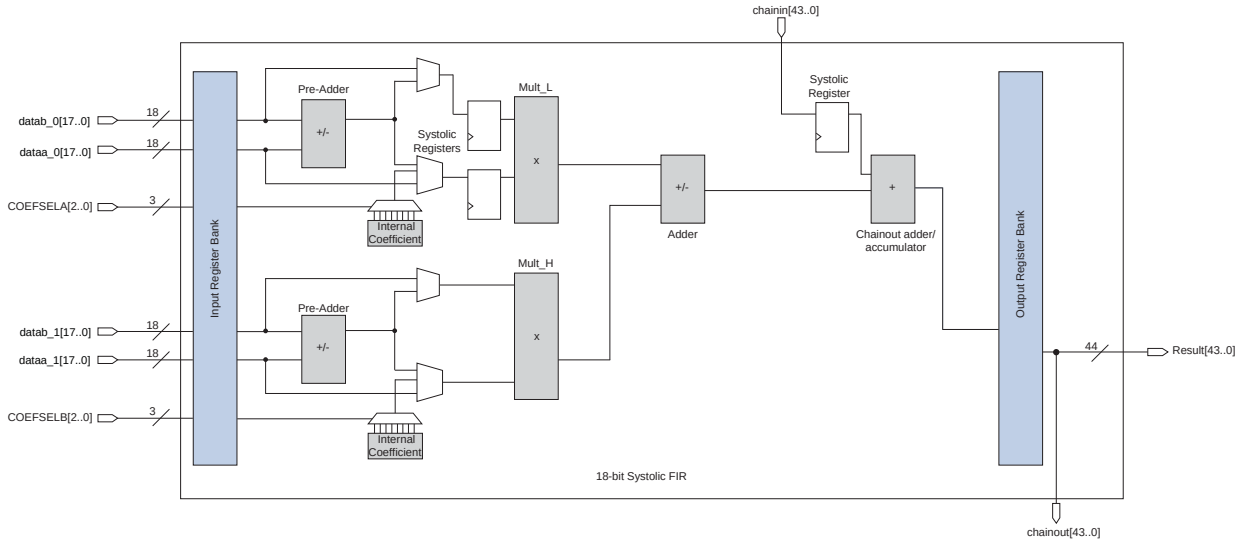
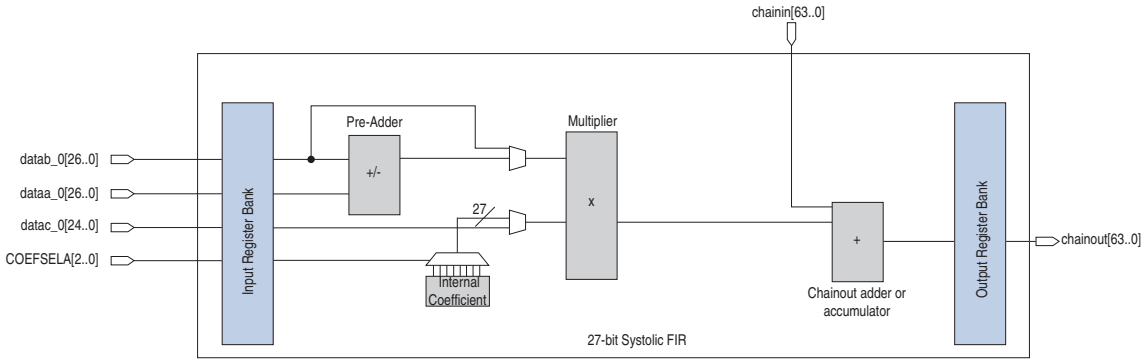


Figure 3–22 shows the 27-bit systolic FIR. This mode allows the implementation of one stage systolic filter per DSP block. The chainout adder or accumulator is configured for a 64-bit operation, providing 10 bits of overhead when using a 27-bit data (54-bit products). This allows a total sum of 1,024 multiplier products.

Figure 3–22. 27-bit Systolic FIR Mode for Stratix V Devices



Variable Precision DSP Block Control Signals

The Stratix V variable precision DSP block has a total of 14 dynamic control signal inputs. The variable precision DSP block dynamic signals are user-configurable and can be set to toggle or not at run time.

Table 3–5 on page 3–22 lists the variable precision DSP block dynamic signals. The Stratix V variable precision DSP block supports 18-bit and 27-bit input cascading.

Table 3–5. Variable Precision DSP Block Dynamic Signals for Stratix V Devices

Signal Name	Function	Count
NEGATE	Control the operation of the decimation	1
LOADCONST	Preload an initial value to the accumulator	1
ACCUMULATE	Enable accumulation	1
SUB	This signal has two functions: ■ Controls add or subtract of the two 18 x 18 multiplier results ■ Controls dynamic switch between 36 x 36 mode and complex 18 x 18	1
COEFSELA COEFSELB	Controls the internal coefficient select multiplexer along with select signals provided through the MSB of each 18-bit data input	2
CLK0 CLK1 CLK2	Variable precision DSP-block-wide clock signals	3
ENA0 ENA1 ENA2	Variable precision DSP-block-wide clock enable signals	3
ACLR0 ACLR1	Variable precision DSP-block-wide asynchronous clear signals	2
Total Count per DSP Block		**14**

Software Support

Altera provides two methods for implementing various modes of the Stratix V variable precision DSP block in a design—using the Quartus II DSP megafunction and HDL inferring.

The following Quartus II megafunctions are supported for the Stratix V variable precision DSP blocks implementation:

- LPM_MULT
- ALTMULT_ADD
- ALTMULT_ACCUM
- ALTMULT_COMPLEX

Alternatively, you can infer the Stratix V variable precision DSP block from the HDL source code. You can use the HDL templates files that are available in the Quartus II software version 11.0 and later to get the Stratix V variable precision DSP blocks inferred. With either method, the Quartus II software maps the functionality to the variable precision DSP blocks during compilation.

For instructions about using the megafunctions and the templates files, refer to the *Quartus II Software Help*.

Document Revision History

Table 3–6 lists the revision history for this chapter.

Table 3–6. Document Revision History

Date	Version	Changes
June 2012	1.4	■ Added Figure 3–2. ■ Updated Figure 3–7, Figure 3–16, and Figure 3–18. ■ Updated Table 3–1. ■ Updated "Chainout Adder and Accumulator" and "18 x 25 Complex Multiplier" sections.
November 2011	1.3	■ Added Figure 3–21. ■ Updated Figure 3–1, Figure 3–2, Figure 3–11, Figure 3–12, Figure 3–14, Figure 3–16, Figure 3–17, Figure 3–18, Figure 3–19, Figure 3–20, and Figure 3–21. ■ Updated Table 3–1 and Table 3–5. ■ Updated "Pre-Adder and Coefficient Select", "Systolic Register", "Systolic FIR Mode", and "Software Support" sections.
May 2011	1.2	■ Updated chapter for Quartus II software 11.0 release. ■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 3–1, Table 3–2, and Table 3–5. ■ Added Table 3–3. ■ Updated all figures in the chapter. ■ Added Figure 3–3. ■ Updated "Software Support" section.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

4. Clock Networks and PLLs in Stratix V Devices

SV51005-1.4

This chapter describes the hierarchical clock networks and phase-locked loops (PLLs) that have advanced features in Stratix® V devices. It includes information about reconfiguring the PLL counter, clock frequency, and phase shift in real time, which allows you to sweep PLL output frequencies and dynamically adjust the output clock phase shift. The Quartus® II software enables the PLLs and their features without external devices.

The chapter contains the following sections:

- "Clock Networks in Stratix V Devices" on page 4–1
- "Stratix V PLLs" on page 4–16

Clock Networks in Stratix V Devices

The global clock networks (GCLKs), regional clock networks (RCLKs), and periphery clock networks (PCLKs) available in Stratix V devices are organized into hierarchical clock structures. The clock networks provide up to 450 unique clock domains (16 GCLKs + 92 RCLKs + 342 PCLKs) within the Stratix V device and allow up to 122 unique GCLK, RCLK, and PCLK clock sources (16 GCLKs + 23 RCLKs + 83 PCLKs) per device quadrant.

Table 4–1 lists the clock resources available in Stratix V devices.

Table 4–1. Clock Resources in Stratix V Devices—*Preliminary*

Clock Resource	Number of Resources Available	Source of Clock Resource
Clock input pins	48 Single-ended (24 Differential)	CLK[0..23]p and CLK[0..23]n pins
GCLK networks	16	CLK[0..23]p and CLK[0..23]n pins, PLL clock outputs, and logic array
RCLK networks	92	CLK[0..23]p and CLK[0..23]n pins, PLL clock outputs, and logic array
PCLK networks	210, 282, 306, and 342 (1)	DPA clock outputs, PLD-transceiver interface clocks, I/O pins, and logic array
GCLKs and RCLKs per quadrant	39	16 GCLKs + 23 RCLKs
GCLKs and RCLKs per device	108	16 GCLKs + 92 RCLKs

Note to Table 4–1:

(1) There are 210 PCLKs in 5SGSD3, 5SGSD4, and 5SGXA3 (with 24 transceivers), 282 PCLKs in 5SGXA3 (with 36 transceivers), 5SGXA4, 5SGSD5, 5SGXB5, and 5SGXB6 devices, 306 PCLKS in 5SGXA5, 5SGXA7, 5SGTC5, 5SGTC7, 5SGSD6, and 5SGSD8 devices, and 342 PCLKs in 5SGXA9, 5SGXAB, 5SGXB9, and 5SGXBB devices.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Stratix V devices have up to 48 dedicated single-ended clock pins or 24 dedicated differential clock pins (CLK[0..23]p and CLK[0..23]n) that can drive either the GCLK or RCLK networks. Table 4–2 on page 4–10 and Table 4–3 on page 4–11 list the clock input pins connectivity to the GCLK and RCLK networks, respectively.

Internally-generated GCLKs, RCLKs, or PCLKs cannot drive the Stratix V PLLs. The input clock to the PLL must be driven by dedicated clock input pins, PLL-fed GCLKs, or PLL-fed RCLKs.

When used as single-ended clock inputs, the CLKn pins drive the PLLs over global or regional clock networks. The CLKn pins do not have dedicated routing paths to the PLLs.

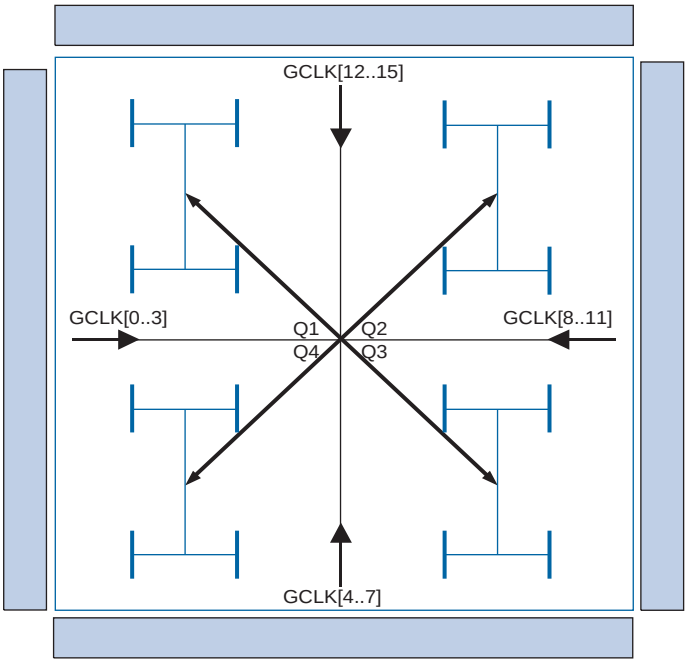
For more information about how to connect the clock input pins, refer to the *Stratix V Device Family Pin Connection Guidelines*.

Global Clock Networks

Stratix V devices provide up to 16 GCLKs that can drive throughout the device, serving as low-skew clock sources for functional blocks such as adaptive logic modules (ALMs), digital signal processing (DSP) blocks, embedded memory blocks, and PLLs. Stratix V device I/O elements (IOEs) and internal logic can also drive GCLKs to create internally generated global clocks and other high fan-out control signals; for example, synchronous or asynchronous clears and clock enables.

Figure 4–1 shows the GCLK networks in Stratix V devices.

Figure 4–1. GCLK Networks

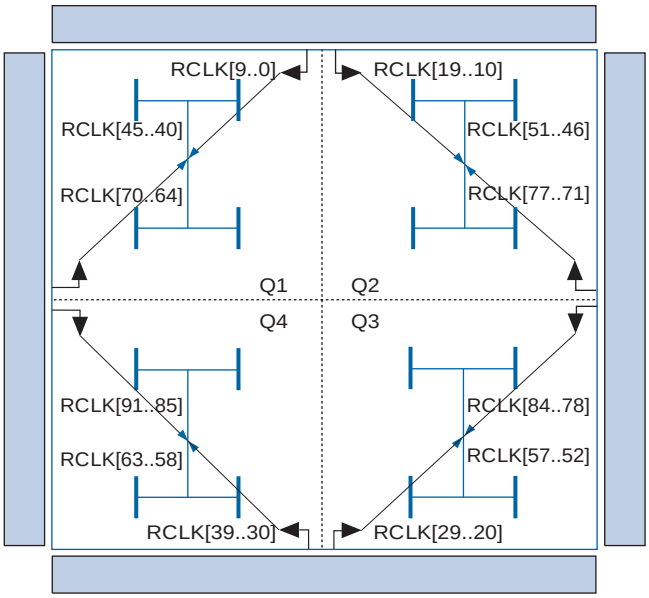


Regional Clock Networks

RCLK networks only pertain to the quadrant they drive into. RCLK networks provide the lowest clock delay and skew for logic contained within a single device quadrant. The Stratix V device IOEs and internal logic within a given quadrant can also drive RCLKs to create internally generated regional clocks and other high fan-out control signals; for example, synchronous or asynchronous clears and clock enables.

Figure 4–2 shows the RCLK networks in Stratix V devices.

Figure 4–2. RCLK Networks



Periphery Clock Networks

The PCLK networks shown in Figure 4–3 through Figure 4–8 on page 4–6 are collections of individual clock networks driven from the periphery of the Stratix V device. Depending on the routing direction, there are vertical PCLKs from the top and bottom periphery and horizontal PCLKs from the left and right periphery. Clock outputs from the dynamic phase aligner (DPA) block, programmable logic device (PLD)-transceiver interface clocks, I/O pins, and internal logic can drive the PCLK networks.

PCLKs have higher skew when compared with GCLK and RCLK networks. You can use PCLKs for general purpose routing to drive signals into and out of the Stratix V device.

Legal clock sources for PCLK networks are clock outputs from the DPA block, PLD-transceiver interface clocks, horizontal I/O pins, and internal logic.

Figure 4–3. PCLK Networks—5SGXA3 (with 36 transceivers), 5SGXA4, and 5SGSD5 Devices

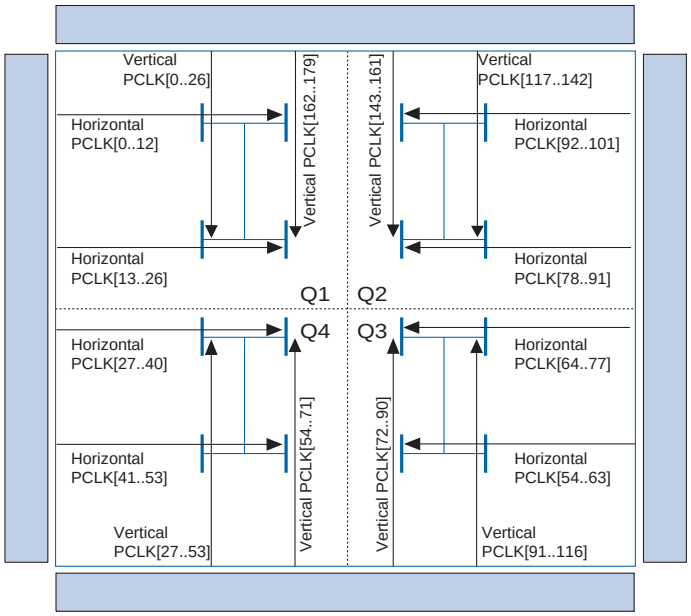


Figure 4–4. PCLK Networks—5SGXB5 and 5SGXB6 Devices

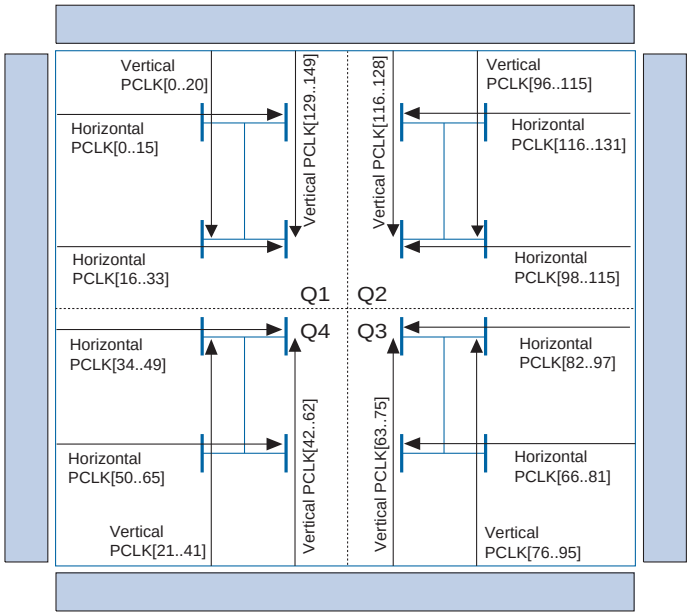


Figure 4–5. PCLK Networks—5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 Devices

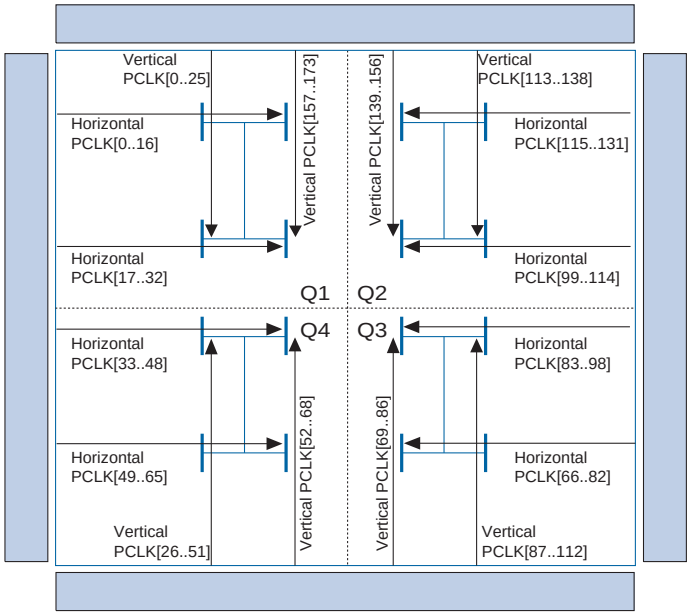


Figure 4–6. PCLK Networks—5SGSD3, 5SGSD4, and 5SGXA3 (with 24 transceivers) Devices

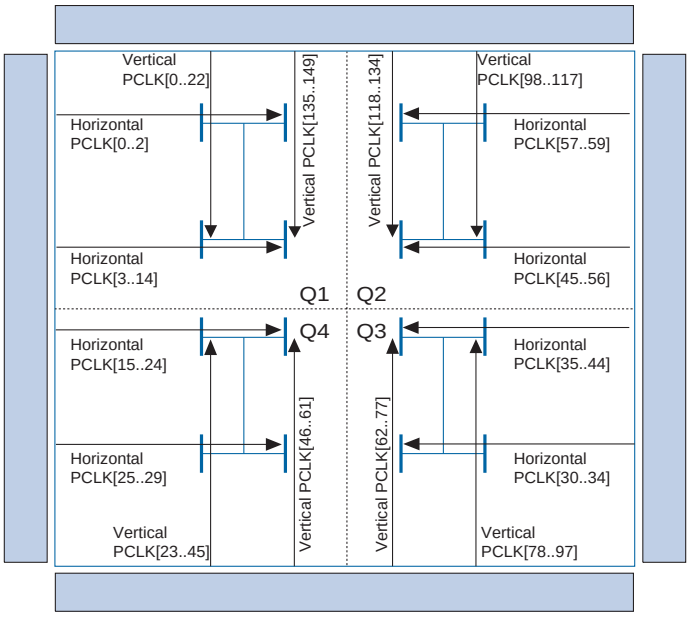


Figure 4–7. PCLK Networks—5SGSD6 and 5SGSD8 Devices

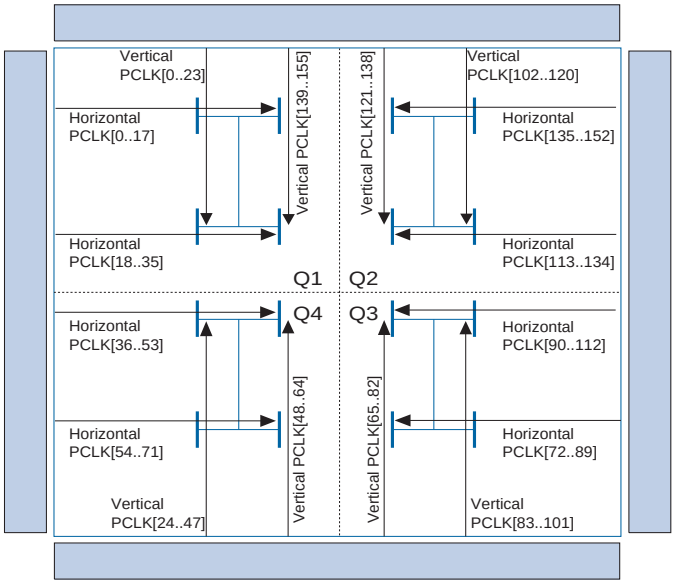
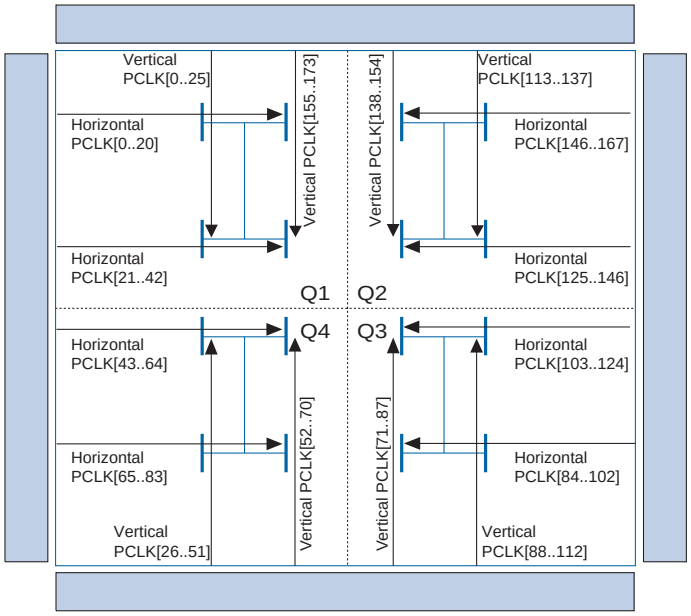


Figure 4–8. PCLK Networks—5SGXA9, 5SGXAB, 5SGXBB, and 5SGXB9 Devices

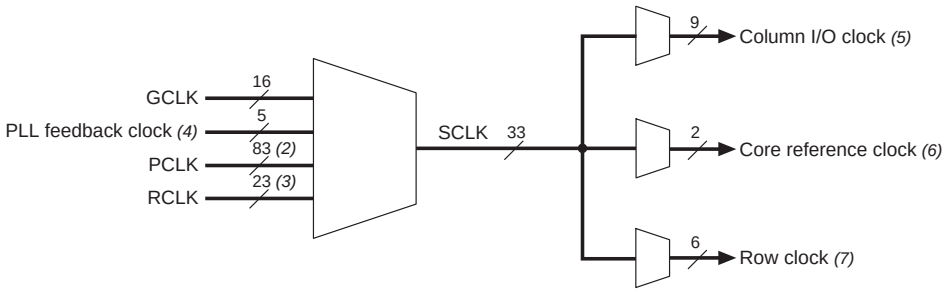


Clock Sources Per Quadrant

There are 33 section clock (SCLK) networks available in each spine clock that can drive six row clocks in each logic array block (LAB) row, nine column I/O clocks, and two core reference clocks. The SCLKs are the clock resources to the core functional blocks, PLLs, and I/O interfaces of the device. Figure 4–9 shows SCLKs driven by the GCLK, RCLK, PCLK, or the PLL feedback clock networks in each spine clock.

A spine clock is another layer of routing between the GCLKs, RCLKs, and PCLK networks before each clock is connected to the clock routing for each LAB row. The settings for spine clocks are transparent. The Quartus II software automatically routes the spine clock based on the GCLK, RCLK, and PCLK networks.

Figure 4–9. Hierarchical Clock Networks Per Spine Clock *(1)*



Notes to Figure 4–9:

(1) The GCLK, RCLK, PCLK, and PLL feedback clocks share the same routing to the SCLKs. The total number of clock resources must not exceed the SCLK limits in each region to ensure successful design fitting in the Quartus II software.
(2) There are up to 83 PCLKs that can drive the SCLKs in each spine clock in the largest device.
(3) There are up to 23 RCLKs that can drive the SCLKs in each spine clock in the largest device.
(4) The PLL feedback clock is the clock from the PLL that drives into the SCLKs.
(5) The column I/O clock is the clock that drives the column I/O core registers and I/O interfaces.
(6) The core reference clock is the clock that feeds into the PLL as the PLL reference clock.
(7) The row clock is the clock source to the LAB, memory blocks, and row I/O interfaces in the core row.

Clock Regions

Stratix V devices provide the following types of clock regions:

- "Entire Device Clock Region"
- "Regional Clock Region"
- "Dual-Regional Clock Region"

Entire Device Clock Region

To form the entire device clock region, a source (not necessarily a clock signal) drives a GCLK network that can be routed through the entire device. This clock region has the maximum delay when compared with other clock regions, but allows the signal to reach every destination within the device. This is a good option for routing global reset and clear signals or routing clocks throughout the device.

Regional Clock Region

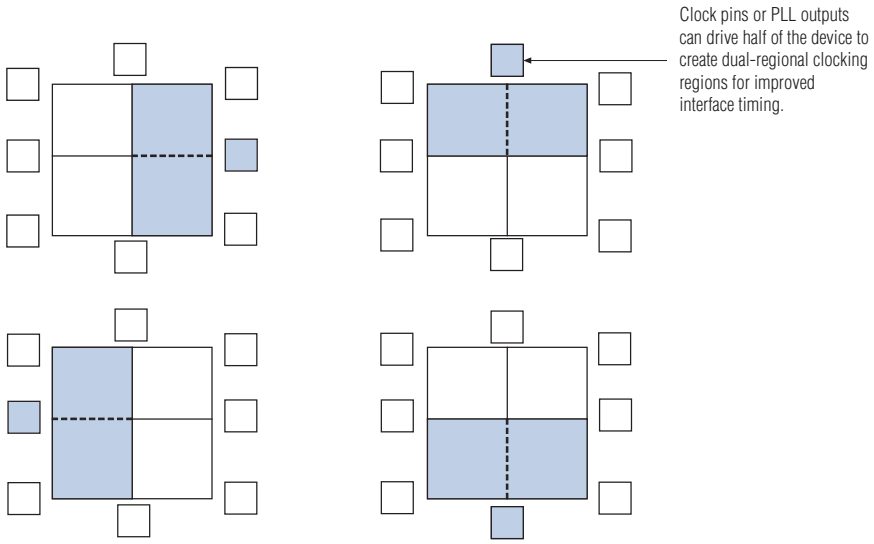
To form a RCLK region, a source drives a single quadrant of the device. This clock region provides the lowest skew within a quadrant and is a good option if all the destinations are within a single device quadrant.

Dual-Regional Clock Region

To form a dual-regional clock region, a single source (a clock pin or PLL output) generates a dual-regional clock by driving two RCLK networks (one from each quadrant). This technique allows destinations across two device quadrants to use the same low-skew clock. The routing of this signal on an entire side has approximately the same delay as a RCLK region. Internal logic can also drive a dual-regional clock network. Corner PLL outputs only span one quadrant, they cannot generate a dual-regional clock network.

Figure 4–10 shows the dual-regional clock region.

Figure 4–10. Dual-Regional Clock Region for Stratix V Devices


Clock pins or PLL outputs can drive half of the device to create dual-regional clocking regions for improved interface timing.

Clock Network Sources

In Stratix V devices, clock input pins, PLL outputs, high-speed serial interface (HSSI) outputs, DPA outputs, and internal logic can drive the GCLK and RCLK networks. For connectivity between the dedicated clock pins, GCLK, and RCLK networks, refer to Table 4–2 and Table 4–3 on page 4–11.

Dedicated Clock Input Pins

CLK pins can be either differential clocks or single-ended clocks. Stratix V devices support up to 24 differential clock inputs or 48 single-ended clock inputs. You can also use dedicated clock input pins CLK[23..0] for high fan-out control signals such as asynchronous clears, presets, and clock enables for protocol signals through the GCLK or RCLK networks. When used as single-ended clock inputs, the CLKn pins drive PLLs over global or regional clock networks.

Internal Logic

You can drive each GCLK, RCLK, and horizontal PCLK network using LAB-routing and row clock to enable internal logic to drive a high fan-out, low-skew signal.

Stratix V PLLs cannot be driven by internally generated GCLKs, RCLKs, or horizontal PCLKs. The input clock to the PLL has to come from dedicated clock input pins or pin/PLL-fed GCLKs or RCLKs.

DPA Outputs

Every DPA generates one PCLK to the core.

HSSI Outputs

Every three HSSI outputs generate a group of six PCLKs to the core.

For more information about DPA and HSSI outputs, refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix V Devices* chapter.

PLL Clock Outputs

Stratix V PLL clock outputs can drive both GCLK and RCLK networks.

Clock Input Pin Connections to GCLK and RCLK Networks

Table 4–2 lists the connection between the dedicated clock input pins and GCLKs.

Table 4–2. Clock Input Pin Connectivity to the GCLK Networks—*Preliminary*

Clock Resources	CLK (p/n Pins)																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
GCLK0	Y	Y	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	Y	Y
GCLK1	Y	Y	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	Y	Y
GCLK2	Y	Y	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	Y	Y
GCLK3	Y	Y	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	Y	Y
GCLK4	—	—	—	—	Y	Y	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
GCLK5	—	—	—	—	Y	Y	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
GCLK6	—	—	—	—	Y	Y	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
GCLK7	—	—	—	—	Y	Y	Y	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
GCLK8	—	—	—	—	—	—	—	—	Y	Y	Y	Y	Y	Y	Y	Y	—	—	—	—	—	—	—	—
GCLK9	—	—	—	—	—	—	—	—	Y	Y	Y	Y	Y	Y	Y	Y	—	—	—	—	—	—	—	—
GCLK10	—	—	—	—	—	—	—	—	Y	Y	Y	Y	Y	Y	Y	Y	—	—	—	—	—	—	—	—
GCLK11	—	—	—	—	—	—	—	—	Y	Y	Y	Y	Y	Y	Y	Y	—	—	—	—	—	—	—	—
GCLK12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	Y	Y	—	—	—	—
GCLK13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	Y	Y	—	—	—	—
GCLK14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	Y	Y	—	—	—	—
GCLK15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	Y	Y	Y	—	—	—	—

Table 4–3 lists the connectivity between the dedicated clock input pins and RCLKs in Stratix V devices. A given clock input pin can drive two adjacent RCLK networks to create a dual-regional clock network.

Table 4–3. Clock Input Pin Connectivity to the RCLK Networks —*Preliminary*

Clock Resources	CLK (p/n pins)																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
RCLK [58,59,60,61,62,63,64,68,85,89]	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [58,59,60,61,62,63,65,69,86,90]	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [58,59,60,61,62,63,66,70,87,91]	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [58,59,60,61,62,63,67,88]	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [20,24,28,30,34,38]	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [21,25,29,31,35,39]	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [22,26,32,36]	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [23,27,33,37]	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [52,53,54,55,56,57,71,75,78,82]	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [52,53,54,55,56,57,72,76,79,83]	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [52,53,54,55,56,57,73,77,80,84]	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [52,53,54,55,56,57,74,81]	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [46,47,48,49,50,51,71,75,78,82]	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—	—
RCLK [46,47,48,49,50,51,72,76,79,83]	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—	—
RCLK [46,47,48,49,50,51,73,77,80,84]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—	—
RCLK [46,47,48,49,50,51,74,81]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—	—
RCLK [0,4,8,10,14,18]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—
RCLK [1,5,9,11,15,19]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—	—
RCLK [2,6,12,16]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—	—
RCLK [3,7,13,17]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—	—
RCLK [40,41,42,43,44,45,64,68,85,89]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—	—
RCLK [40,41,42,43,44,45,65,69,86,90]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—	—
RCLK [40,41,42,43,44,45,66,70,87,91]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y	—
RCLK [40,41,42,43,44,45,67,88]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Y

Clock Output Connections

For Stratix V PLL connectivity to GCLK and RCLK networks, refer to *PLL Connectivity to GCLK and RCLK Networks for Stratix V Devices*.

Clock Control Block

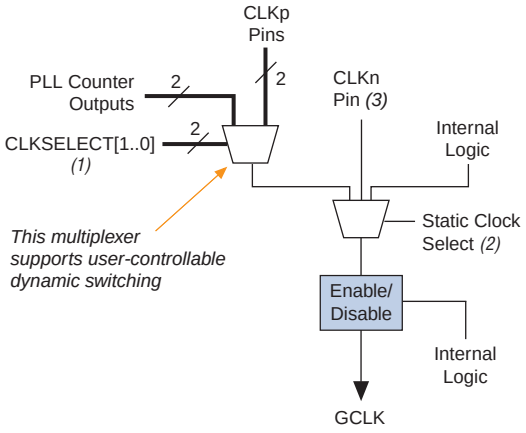
Every GCLK, RCLK, and PCLK network has its own clock control block. The control block provides the following features:

- Clock source selection (dynamic selection available only for GCLKs)
- Global clock multiplexing
- Clock power down (static or dynamic clock enable or disable available only for GCLKs and RCLKs)

Figure 4–11, Figure 4–12, and Figure 4–13 show the GCLK, RCLK, and PCLK control blocks, respectively.

You can select the clock source for the GCLK select block either statically or dynamically. You can statically select the clock source using a setting in the Quartus II software or you can dynamically select the clock source using internal logic to drive the multiplexer-select inputs. When selecting the clock source dynamically, you can select either PLL outputs (such as `C0` or `C1`) or a combination of clock pins or PLL outputs.

Figure 4–11. GCLK Control Block for Stratix V Devices



Notes to Figure 4–11:

(1) When the device is in user mode, you can dynamically control the clock select signals through internal logic.

(2) When the device is in user mode, you can only set the clock select signals through a configuration file (SRAM object file [**.sof**] or programmer object file [**.pof**]); they cannot be dynamically controlled.

(3) The `CLKn` pin is not a dedicated clock input when used as a single-ended PLL clock input. The `CLKn` pin can drive the PLL using the GCLK.

You can set the input clock sources and the `clkena` signals for the GCLK and RCLK network multiplexers through the Quartus II software using the ALTCLKCTRL megafunction.

When using the ALTCLKCTRL megafunction to implement dynamic clock source selection, the inputs from the clock pins feed the `inclk[0..1]` ports of the multiplexer, while the PLL outputs feed the `inclk[2..3]` ports. You can choose from among these inputs using the `CLKSELECT[1..0]` signal.

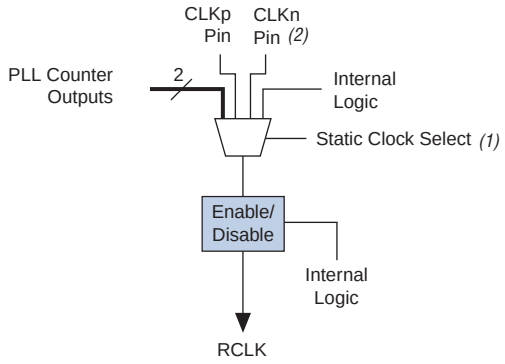
For more information about using the ALTCLKCTRL megafunction for dynamic clock source selection, refer to the *Clock Control Block (ALTCLKCTRL) Megafunction User Guide*.

The mapping between the input clock pins, PLL counter outputs, and clock control block inputs is as follows:

- `inclk[0]` and `inclk[1]`—can be fed by any of the four dedicated clock pins on the same side of the Stratix V device
- `inclk[2]`—can be fed by PLL counters C0 and C2 from the two center PLLs on the same side of the Stratix V device
- `inclk[3]`—can be fed by PLL counters C1 and C3 from the two center PLLs on the same side of the Stratix V device

Corner PLLs cannot be used for dynamic clock control selection.

Figure 4–12. RCLK Control Block



Notes to Figure 4–12:

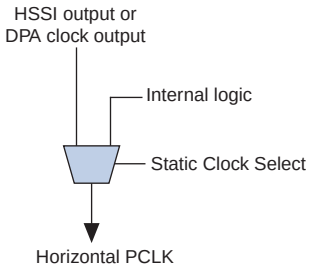
(1) When the device is in user mode, you can only set the clock select signals through a configuration file (**.sof** or **.pof**); they cannot be dynamically controlled.

(2) The `CLKn` pin is not a dedicated clock input when used as a single-ended PLL clock input. The `CLKn` pin can drive the PLL using the RCLK.

You can only control the clock source selection for the RCLK select block statically using configuration bit settings in the configuration file (**.sof** or **.pof**) generated by the Quartus II software.

You can select the HSSI output or internal logic to drive the HSSI horizontal PCLK control block. Alternatively, you can also use the DPA clock output or internal logic to drive the DPA horizontal PCLK. You can only use the DPA output to generate the vertical PCLK to the core.

Figure 4–13. Horizontal PCLK Control Block



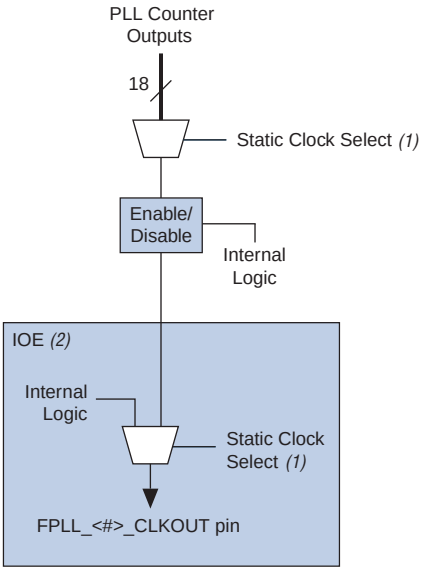
You can power down the Stratix V GCLK and RCLK clock networks using both static and dynamic approaches. When a clock network is powered down, all the logic fed by the clock network is in off-state, thereby reducing the overall power consumption of the device. The unused GCLK, RCLK, and PCLK networks are automatically powered down through configuration bit settings in the configuration file (**.sof** or **.pof**) generated by the Quartus II software.

The dynamic clock enable or disable feature allows the internal logic to control power-up or power-down synchronously on the GCLK and RCLK networks, including dual-regional clock regions. Figure 4–11 and Figure 4–12 show that this function is independent of the PLL and is applied directly on the clock network.

☞ You cannot dynamically enable or disable GCLK or RCLK networks that drive PLLs.

You can enable or disable the dedicated external clock output pins using the ALTCLKCTRL megafunction. Figure 4–14 shows the external PLL output clock control block.

Figure 4–14. External PLL Output Clock Control Block for Stratix V Devices



Notes to Figure 4–14:

(1) When the device is in user mode, you can only set the clock select signals through a configuration file (**.sof** or **.pof**); they cannot be dynamically controlled.

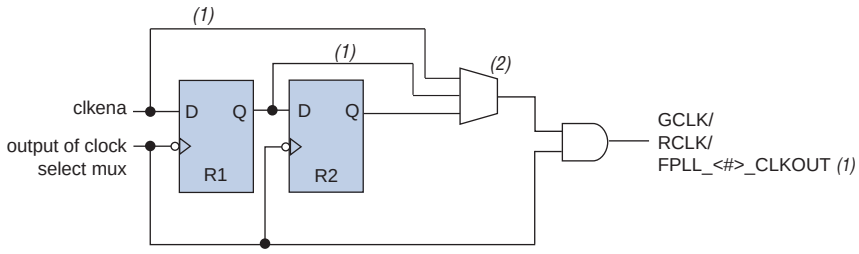
(2) The clock control block feeds to a multiplexer within the FPLL_<#>_CLKOUT pin's IOE. The FPLL_<#>_CLKOUT pin is a dual-purpose pin. Therefore, this multiplexer selects either an internal signal or the output of the clock control block.

Clock Enable Signals

Figure 4–15 shows how the clock enable and disable circuit of the clock control block is implemented in Stratix V devices.

You cannot use the clock enable and disable circuit of the clock control block if the GCLK or RCLK output drives the input of a PLL.

Figure 4–15. clkena Implementation



Notes to Figure 4–15:

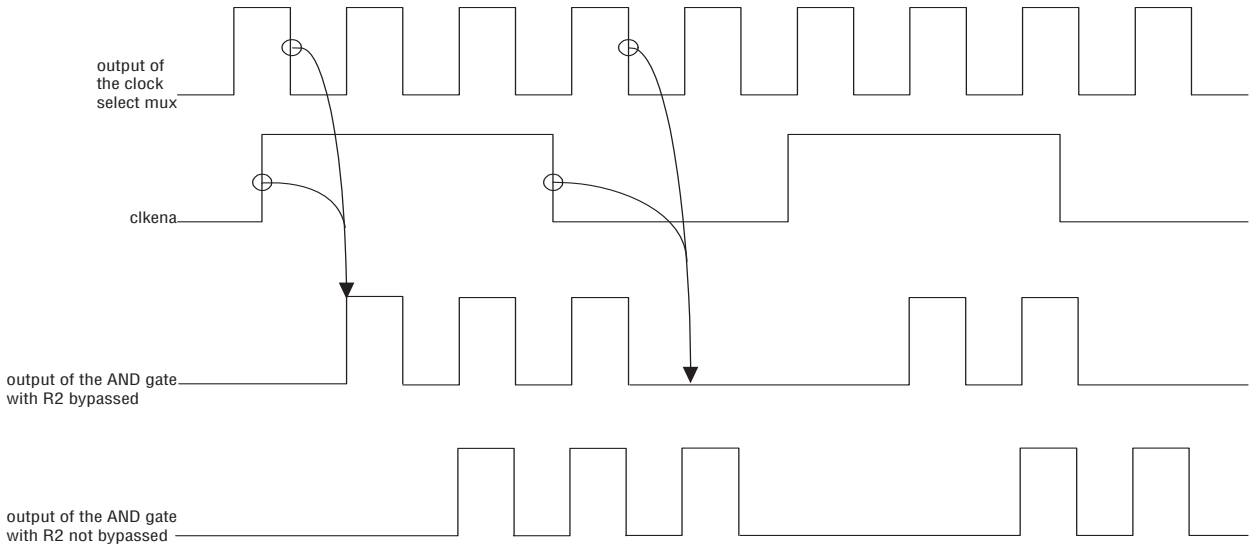
(1) The R1 and R2 bypass paths are not available for the PLL external clock outputs.

(2) The select line is statically controlled by a bit setting in the configuration file (**.sof** or **.pof**).

In Stratix V devices, the clkena signals are supported at the clock network level instead of at the PLL output counter level. This allows you to gate off the clock even when you are not using a PLL. You can also use the clkena signals to control the dedicated external clocks from the PLLs. Figure 4–16 shows a waveform example for a clock output enable. clkena is synchronous to the falling edge of the clock output.

Stratix V devices also have an additional metastability register that aids in asynchronous enable and disable of the GCLK and RCLK networks. You can optionally bypass this register in the Quartus II software.

Figure 4–16. clkena Signals *(1)*



Note to Figure 4–16:

(1) You can use the clkena signals to enable or disable the GCLK and RCLK networks or the FPLL_<#>_CLKOUT pins.

The PLL can remain locked independent of the clkena signals because the loop-related counters are not affected. This feature is useful for applications that require a low-power or sleep mode. The clkena signal can also disable clock outputs if the system is not tolerant of frequency overshoot during resynchronization.

Stratix V PLLs

Stratix V device family introduces the fractional PLLs in addition to the existing integer PLLs that provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces. Each fractional PLL has 18 output counters that supports integer or fractional frequency synthesis.

Stratix V devices offer up to 32 fractional PLLs in the larger densities. All Stratix V fractional PLLs have the same core analog structure and features support.

Table 4–4 lists the features of PLLs in Stratix V devices.

Table 4–4. PLL Features for Stratix V Devices —*Preliminary*

Feature	Stratix V
Integer PLL	Yes
Fractional PLL	Yes
C output counters	18
M, N, C counter sizes	1 to 512
Dedicated external clock outputs	4 single-ended or 2 single-ended and 1 differential
Dedicated clock input pins	4 single-ended or 4 differential
External feedback input pin	Single-ended or differential
Spread-spectrum input clock tracking	Yes *(1)*
Source synchronous compensation	Yes
Direct compensation	Yes
Normal compensation	Yes
Zero delay buffer (ZDB) compensation	Yes
External feedback compensation	Yes
LVDS compensation	Yes
VCO output drives the DPA clock	Yes
Phase shift resolution	78.125 ps *(2)*
Programmable duty cycle	Yes

Notes to Table 4–4:

(1) Provided input clock jitter is within input jitter tolerance specifications.

(2) The smallest phase shift is determined by the voltage-controlled oscillator (VCO) period divided by eight. For degree increments, the Stratix V device can shift all output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the frequency and divide parameters.

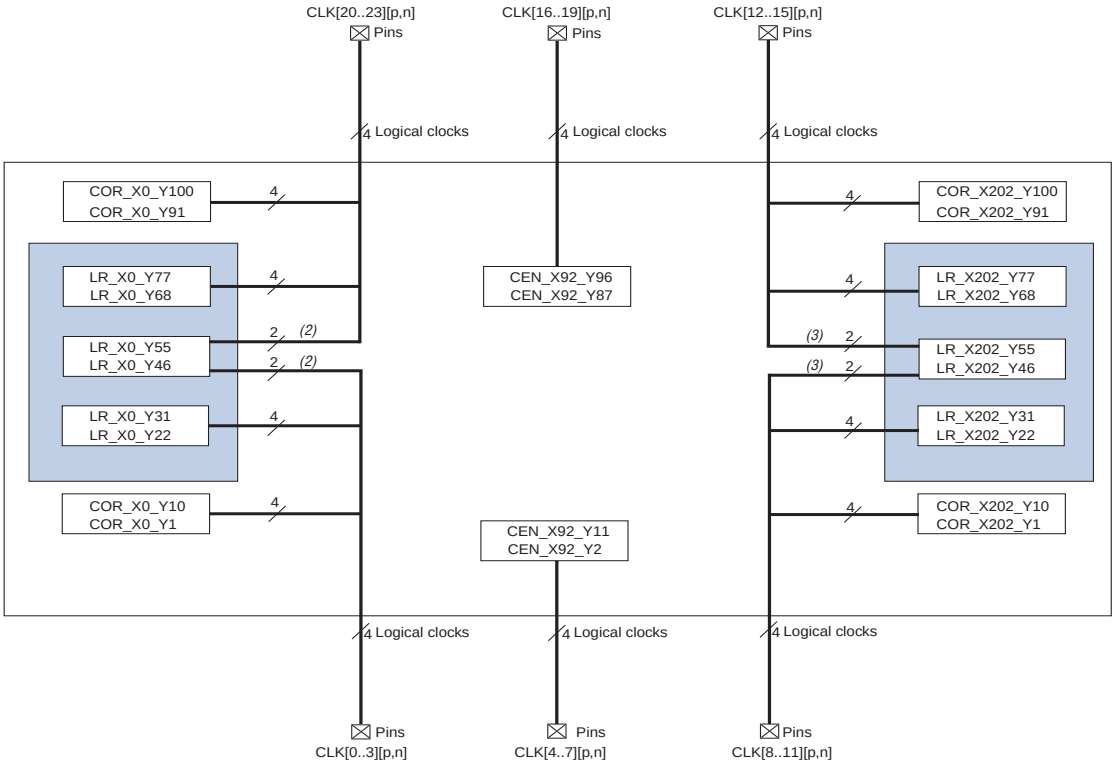
Figure 4–17 through Figure 4–22 on page 4–23 show the physical locations of the fractional PLLs. For single-ended clock inputs, only the CLK<#>p pin has a dedicated connection to the PLL. If you use the CLK<#>n pin, a global or regional clock is used.

Driving a PLL over a global or regional clock can lead to higher jitter at the PLL input, and the PLL will not be able to fully compensate for the global or regional clock. Altera® recommends to use the CLK<#>p pin for optimal performance when you use single-ended clock inputs to drive the PLLs.

The nomenclature for Stratix V PLLs follow their geographical coordinates in the device floor plan. The PLLs that reside on the center of the device are named CEN_X<#>_Y<#>, the PLLs that reside on the corner of the device are named COR_X<#>_Y<#>, and the PLLs that reside on the left and right sides of the device are named LR_X<#>_Y<#>.

Figure 4–17 shows the PLL locations for 5SGXA3 (with 36 transceivers), 5SGXA4, and 5SGSD5 devices.

Figure 4–17. PLL Locations for 5SGXA3 (with 36 transceivers), 5SGXA4, and 5SGSD5 Devices *(1)*



Notes to Figure 4–17:

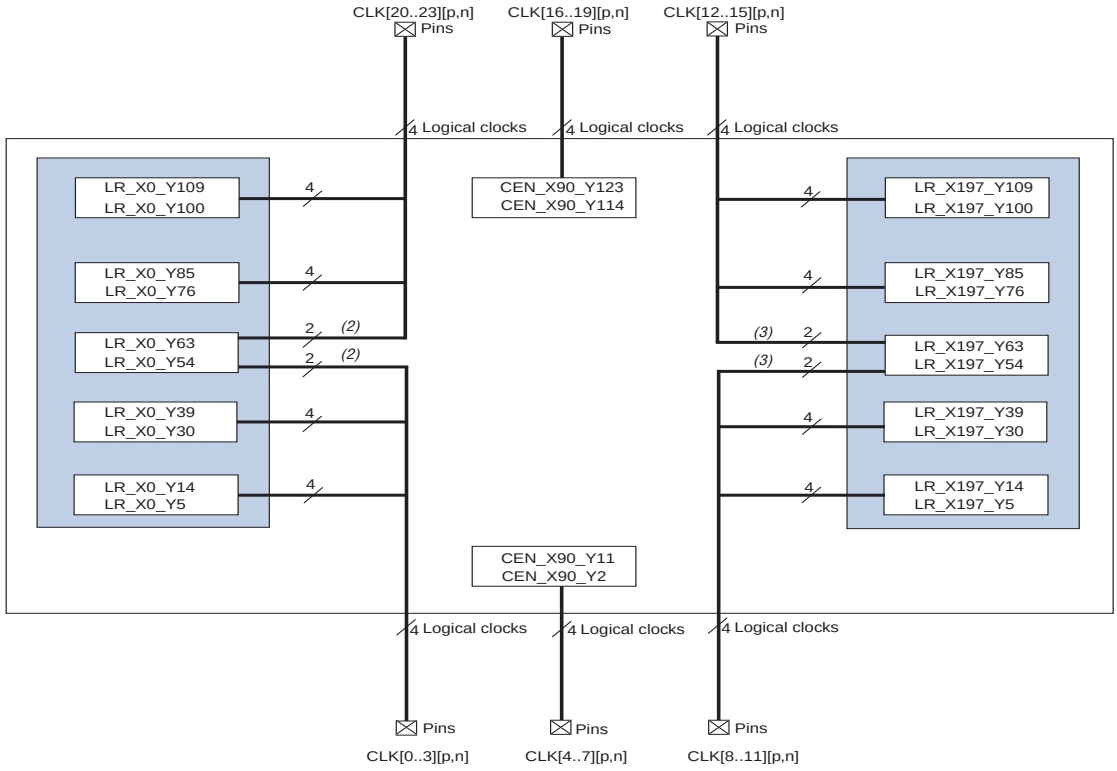
(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

(2) CLK0, CLK1, CLK22, and CLK23 clock pins feed into fractional PLL LR_X0_Y46 and fractional PLL LR_X0_Y55.

(3) CLK8, CLK9, CLK14, and CLK15 clock pins feed into fractional PLL LR_X202_Y46 and fractional PLL LR_X202_Y55.

Figure 4–18 shows the PLL locations for 5SGXB5 and 5SGXB6 devices.

Figure 4–18. PLL Locations for 5SGXB5 and 5SGXB6 Devices *(1)*



Notes to Figure 4–18:

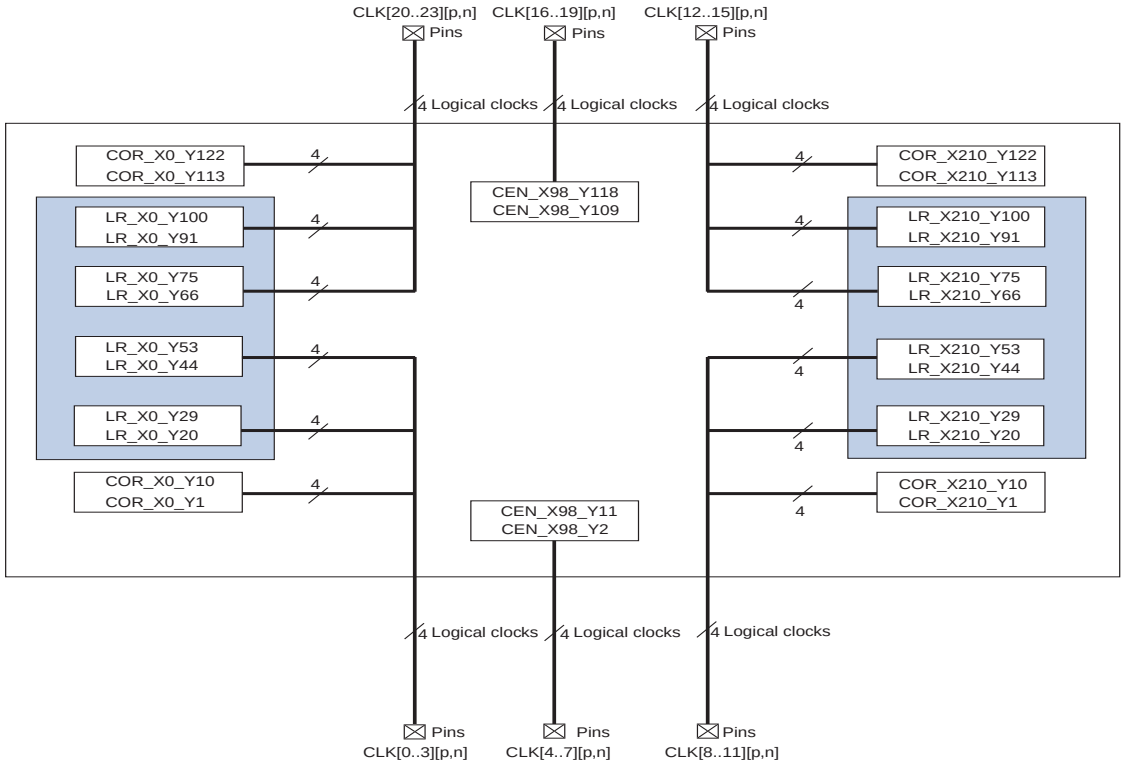
(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

(2) CLK0, CLK1, CLK22, and CLK23 clock pins feed into fractional PLL LR_X0_Y54 and fractional PLL LR_X0_Y63.

(3) CLK8, CLK9, CLK14, and CLK15 clock pins feed into fractional PLL LR_X197_Y54 and fractional PLL LR_X197_Y63.

Figure 4–19 shows the PLL locations for 5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 devices.

Figure 4–19. PLL Locations for 5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 Devices *(1)*

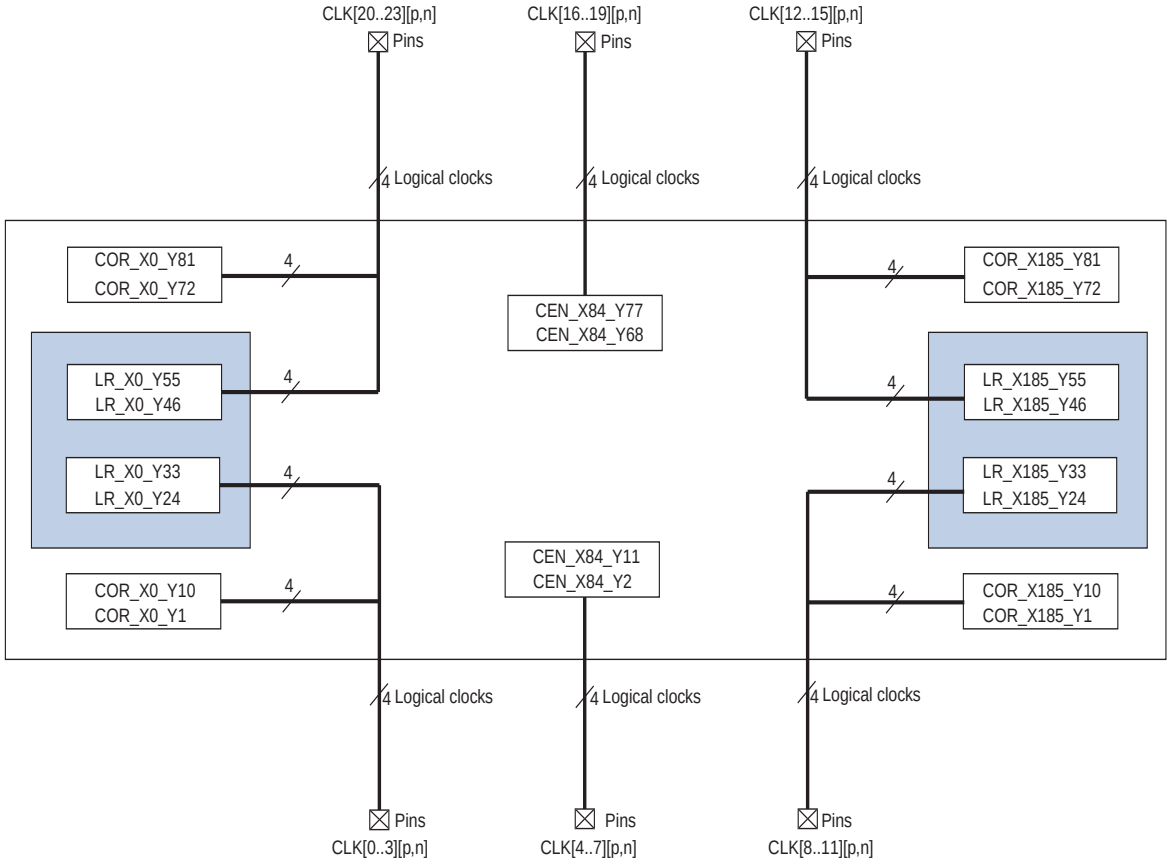


Note to Figure 4–19:

(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

Figure 4–20 shows the PLL locations for 5SGSD3, 5SGSD4, and 5SGXA3 (with 24 transceivers) devices.

Figure 4–20. PLL Locations for 5SGSD3, 5SGSD4, and 5SGXA3 (with 24 transceivers) Devices *(1)*

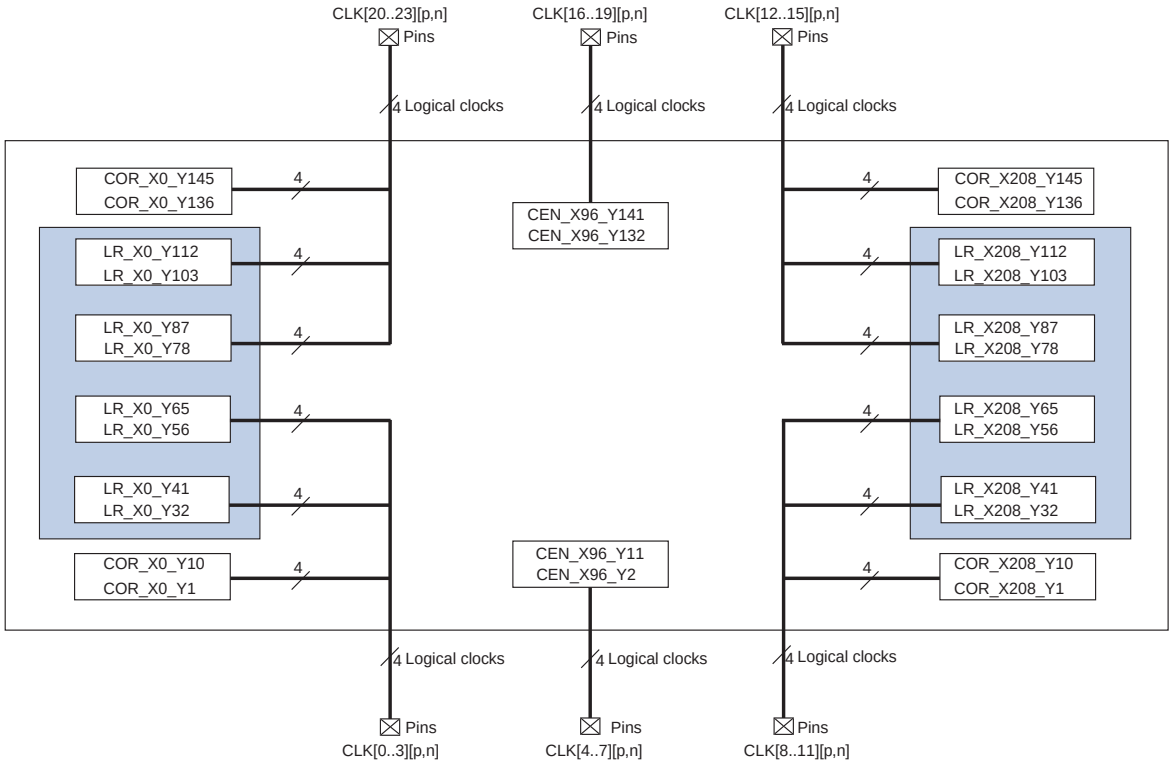


Note to Figure 4–20:

(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

Figure 4–21 shows the PLL locations for 5SGSD6 and 5SGSD8 devices.

Figure 4–21. PLL Locations for 5SGSD6 and 5SGSD8 Devices *(1)*

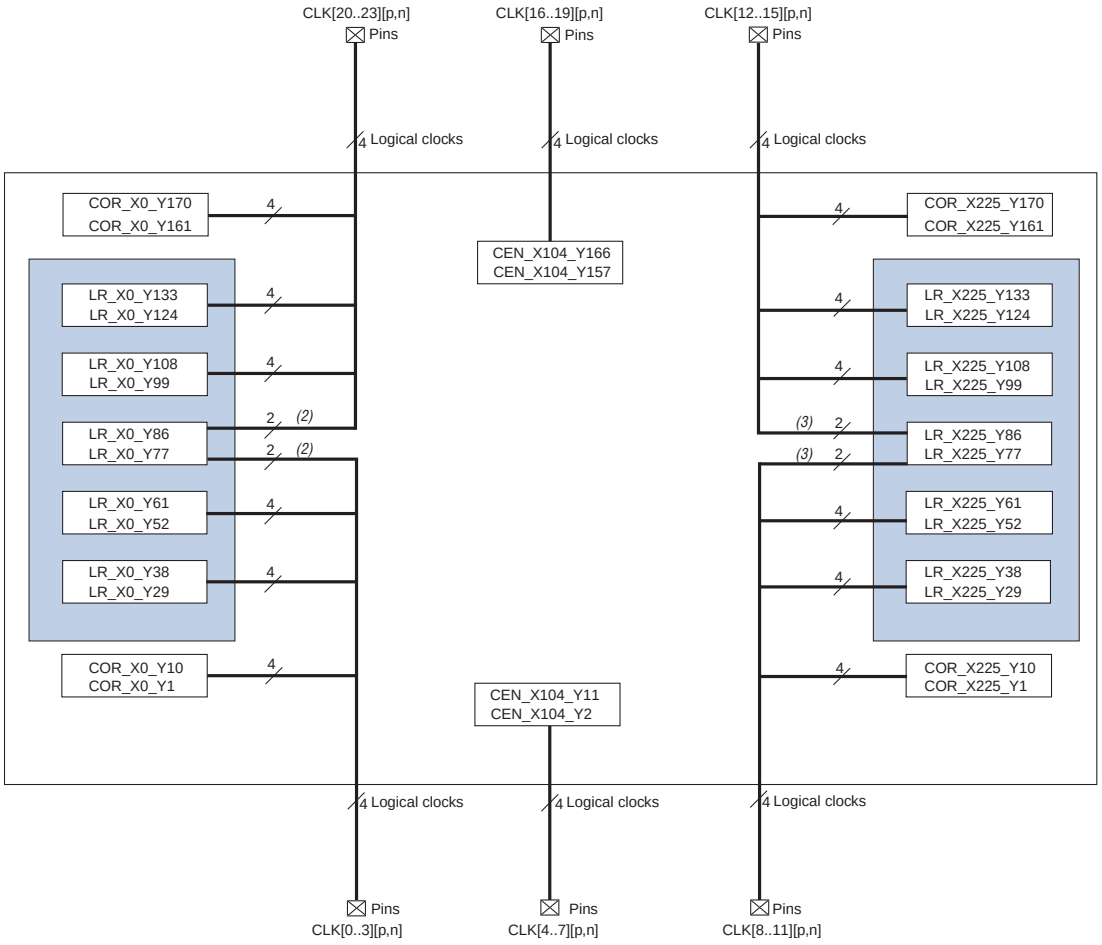


Note to Figure 4–21:

(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

Figure 4–22 shows the PLL locations for 5SEE9, 5SEEB, 5SGXA9, 5SGXAB, 5SGXB9, and 5SGXBB devices.

Figure 4–22. PLL Locations for 5SEE9, 5SEEB, 5SGXA9, 5SGXAB, 5SGXB9, and 5SGXBB Devices *(1)*



Notes to Figure 4–22:

(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

(2) `CLK0`, `CLK1`, `CLK22`, and `CLK23` clock pins feed into fractional PLL `LR_X0_Y77` and fractional PLL `LR_X0_Y86`.

(3) `CLK8`, `CLK9`, `CLK14`, and `CLK15` clock pins feed into fractional PLL `LR_X225_Y77` and fractional PLL `LR_X225_Y86`.

PLL Migration Guidelines

If you plan to migrate your design between 5SGXA5, 5SGXA7, 5SGXA9, 5SGXAB, 5SGXB9, 5SGXBB, 5SGXD6, and 5SGXD8 devices with 48 transceiver channels, and your design requires a PLL to drive the HSSI and clock network (GCLK or RCLK) simultaneously, use the 2 middle PLLs on the left or right side of the device.

Table 4–5 shows the location of the middle PLLs in these devices.

Table 4–5. Location of Middle PLLs

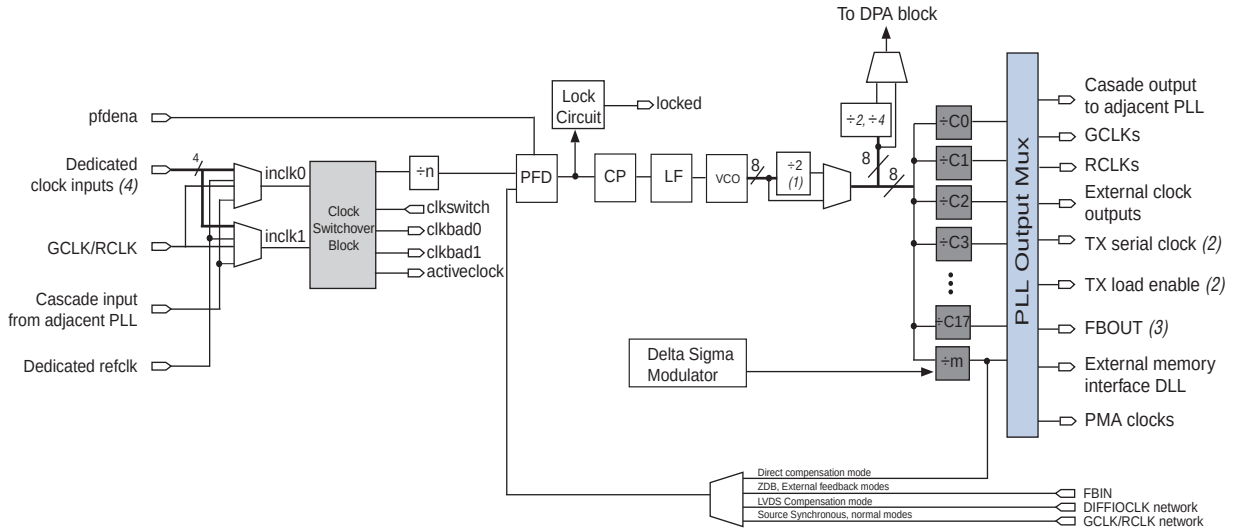
Device	Middle PLL Location	
	Left Side	Right Side
5SGXA5	LR_X0_Y53, LR_X0_Y66	LR_X210_Y53, LR_X210_Y66
5SGXA7		
5SGXA9	LR_X0_Y77, LR_X0_Y86	LR_X225_Y77, LR_X225_Y86
5SGXAB		
5SGXB9		
5SGXBB		
5SGXD6	LR_X0_Y65, LR_X0_Y78	LR_X208_Y65, LR_X208_Y78
5SGXD8		

For the CLKIN pin connectivity to the middle PLLs, refer to Figure 4–17 on page 4–18 to Figure 4–22 on page 4–23.

Fractional PLL Architecture

Figure 4–23 shows the high-level block diagram of the Stratix V fractional PLL.

Figure 4–23. Fractional PLL High-Level Block Diagram



Notes to Figure 4–23:

(1) This is the VCO post-scale counter K.

(2) Only C0, C2, C15, and C17 can drive the TX serial clock and C1, C3, C14, and C16 can drive the TX load enable.

(3) This FBOUT port is fed by the M counter in the Stratix V PLLs.

(4) For single-ended clock inputs, only the CLK<#>p pin has a dedicated connection to the PLL. If you use the CLK<#>n pin, a global or regional clock is used.

Fractional PLL Usage

You can configure the fractional PLL as either an integer or an enhanced fractional mode. One fractional PLL can use up to 18 output counters and all external clock outputs. Two adjacent fractional PLLs share the 18 output counters.

You can use fractional PLLs to reduce the number of oscillators required on the board, as well as to reduce the clock pins used in the FPGA by synthesizing multiple clock frequencies from a single reference clock source. In addition, you can use fractional PLLs for clock network delay compensation, zero-delay buffering, and transmit clocking for transceivers.

PLL External Clock I/O Pins

Two adjacent corner and center fractional PLLs share four dual-purpose clock I/O pins, organized as one of the following combinations:

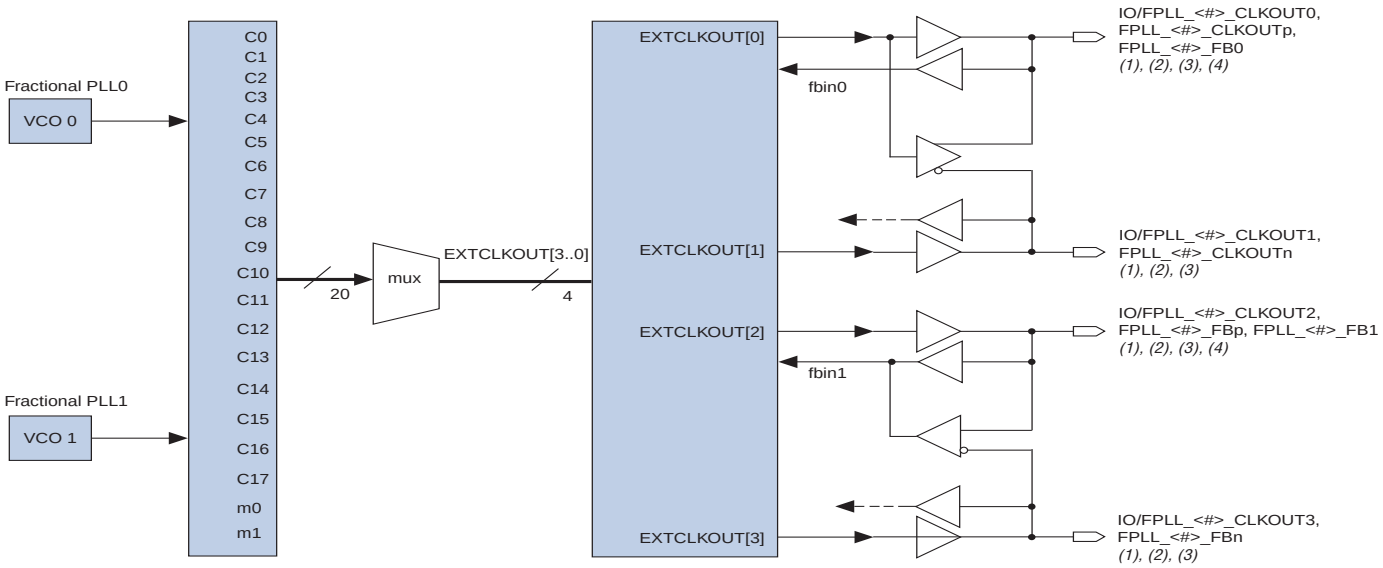
- Four single-ended clock outputs
- Two single-ended outputs and one differential clock output
- Four single-ended clock outputs and two single-ended feedback inputs within the I/O driver feedback for ZDB support
- Two single-ended clock outputs and two single-ended feedback inputs for single-ended External Feedback (EFB) support

- One differential clock output and one differential feedback input for differential EFB support (only one of the two adjacent fractional PLLs can support differential EFB at one time while the other fractional PLL can be used for general-purpose clocking)

All left and right fractional PLLs in Stratix V devices do not support external clock outputs.

Figure 4–24 shows the dual-purpose clock I/O pins associated with the PLL for Stratix V devices.

Figure 4–24. Dual-Purpose Clock I/O Pins Associated with PLL for Stratix V Devices



Notes to Figure 4–24:

(1) You can feed these clock output pins using any one of the C[17..0] or m counters. When not used as external clock outputs, these clock output pins can be used as regular user I/Os.

(2) The FPLL_<#>_CLKOUT0, FPLL_<#>_CLKOUT1, FPLL_<#>_CLKOUT2, and FPLL_<#>_CLKOUT3 pins are single-ended clock output pins.

(3) The FPLL_<#>_CLKOUTp and FPLL_<#>_CLKOUTn pins are differential output pins while the FPLL_<#>_FBp and FPLL_<#>_FBn pins are differential feedback input pins to support differential EFB.

(4) The FPLL_<#>_FB0 and FPLL_<#>_FB1 pins are single-ended feedback input pins.

Figure 4–24 shows that any of the output counters (C[17..0]) on the PLLs or the M counter can feed the dedicated external clock outputs. Therefore, one counter or frequency can drive all output pins available from a given PLL.

Each pin of a single-ended output pair can either be in-phase or 180° out-of-phase. The Quartus II software places the NOT gate in the design into the IOE to implement the 180° phase with respect to the other pin in the pair. The clock output pin pairs support the same I/O standards as standard output pins as well as LVDS, LVPECL, differential high-speed transceiver logic (HSTL), and differential SSTL.

To determine which I/O standards are supported by the PLL clock input and output pins, refer to the *I/O Features in Stratix V Devices* chapter.

Stratix V PLLs can also drive out to any regular I/O pin through the GCLK or RCLK network. You can also use the external clock output pins as user I/O pins if you do not require external PLL clocking.

PLL Control Signals

You can use the `pfdena`, `areset`, and `locked` signals to control and observe PLL operation and resynchronization.

pfdena

Use the `pfdena` signal to maintain the most recent locked frequency so your system has time to store its current settings before shutting down. The `pfdena` signal controls the PFD output with a programmable gate. If you disable PFD, the VCO operates at its most recent set value of control voltage and frequency, with some long-term drift to a lower frequency. The PLL continues running even if it goes out-of-lock or the input clock is disabled. You can use either your own control signal or the control signals available from the clock switchover circuit (`activeclock`, `clkbad[0]`, or `clkbad[1]`) to control `pfdena`.

areset

The `areset` signal is the reset or resynchronization input for each PLL. The device input pins or internal logic can drive these input signals. When `areset` is driven high, the PLL counters reset, clearing the PLL output and placing the PLL out-of-lock. The VCO is then set back to its nominal setting. When `areset` is driven low again, the PLL resynchronizes to its input as it re-locks.

You must assert the `areset` signal every time the PLL loses lock to guarantee the correct phase relationship between the PLL input and output clocks. You can set up the PLL to automatically reset (self reset) after a loss-of-lock condition using the Quartus II MegaWizard™ Plug-In Manager. You must include the `areset` signal if either of the following conditions is true:

- PLL reconfiguration or clock switchover is enabled in the design
- Phase relationships between the PLL input and output clocks must be maintained after a loss-of-lock condition

If the input clock to the PLL is not toggling or is unstable after power up, assert the `areset` signal after the input clock is stable and within specifications.

locked

The `locked` signal output of the PLL indicates that the PLL has locked onto the reference clock and the PLL clock outputs are operating at the desired phase and frequency set in the MegaWizard Plug-In Manager. The lock detection circuit provides a signal to the core logic that gives an indication when the feedback clock has locked onto the reference clock both in phase and frequency.

Altera recommends using the `areset` and `locked` signals in your designs to control and observe the status of your PLL.

Clock Feedback Modes

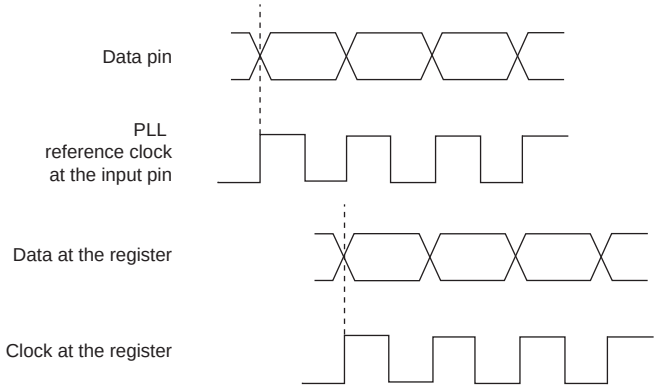
Stratix V PLLs support up to six different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle.

The input and output delays are fully compensated by a PLL only when using the dedicated clock input pins associated with a given PLL as the clock source. When a RCLK or GCLK network drives the PLL or the PLL is driven by a dedicated clock pin that is not associated with the PLL, the input and output delays may not be fully compensated in the Quartus II software. An example is when you configure a PLL in zero-delay buffer mode and the PLL input is driven by an associated dedicated clock input pin. In this configuration, a fully compensated clock path results in zero delay between the clock input and one of the output clocks from the PLL. However, if the PLL input is instead fed by a non-dedicated input (using the GCLK network), the output clock may not be perfectly aligned with the input clock. Refer to Figure 4–17 on page 4–18 through Figure 4–21 on page 4–22 for a mapping of dedicated clock pins to their associated PLLs.

Source Synchronous Mode

If the data and clock arrive at the same time on the input pins, the same phase relationship is maintained at the clock and data ports of any IOE input register. Figure 4–25 shows an example waveform of the clock and data in this mode. Altera recommends source synchronous mode for source-synchronous data transfers. Data and clock signals at the IOE experience similar buffer delays as long as you use the same I/O standard.

Figure 4–25. Phase Relationship Between Clock and Data in Source Synchronous Mode



Source synchronous mode compensates for the delay of the clock network used plus any difference in the delay between these two paths:

- Data pin to the IOE register input
- Clock input pin to the PLL PFD input

The Stratix V PLL can compensate multiple pad-to-input-register paths, such as a data bus when it is set to use source-synchronous compensation mode. You can use the "PLL Compensation" assignment in the Quartus II software Assignment Editor to select which input pins are used as the PLL compensation targets. You can include your entire data bus, provided the input registers are clocked by the same output of a source-synchronous-compensated PLL. In order for the clock delay to be properly compensated, all of the input pins must be on the same side of the device. The PLL compensates for the input pin with the longest pad-to-register delay among all input pins in the compensated bus.

If you do not make the "PLL Compensation" assignment, the Quartus II software automatically selects all of the pins driven by the compensated output of the PLL as the compensation target.

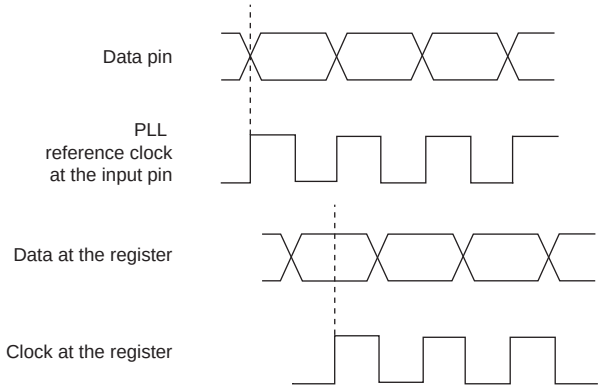
Source Synchronous Mode for LVDS Compensation

The goal of source synchronous mode is to maintain the same data and clock timing relationship seen at the pins of the internal serializer/deserializer (SERDES) capture register, except that the clock is inverted (180° phase shift). Thus, source synchronous mode ideally compensates for the delay of the LVDS clock network plus any difference in delay between these two paths:

- Data pin-to-SERDES capture register
- Clock input pin-to-SERDES capture register. In addition, the output counter must provide the 180° phase shift

Figure 4–26 shows an example waveform of the clock and data in LVDS mode.

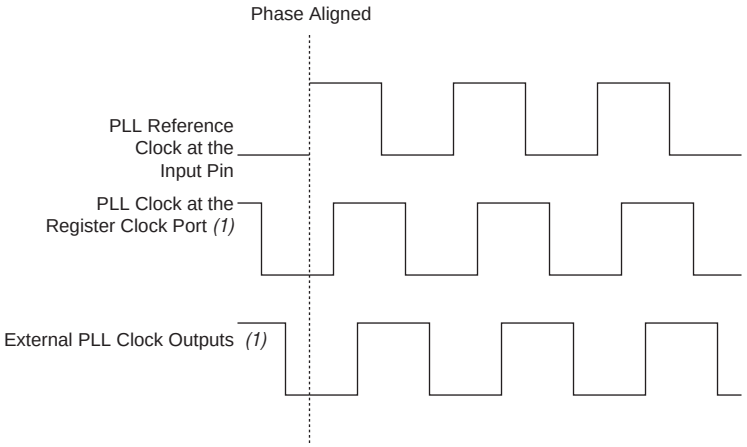
Figure 4–26. Phase Relationship Between the Clock and Data in LVDS Mode



Direct Compensation Mode

In direct compensation mode, the PLL does not compensate for any clock networks. This mode provides better jitter performance because the clock feedback into the PFD passes through less circuitry. Both the PLL internal- and external-clock outputs are phase-shifted with respect to the PLL clock input. Figure 4–27 shows an example waveform of the PLL clocks' phase relationship in direct compensation mode.

Figure 4–27. Phase Relationship Between the PLL Clocks in Direct Compensation Mode



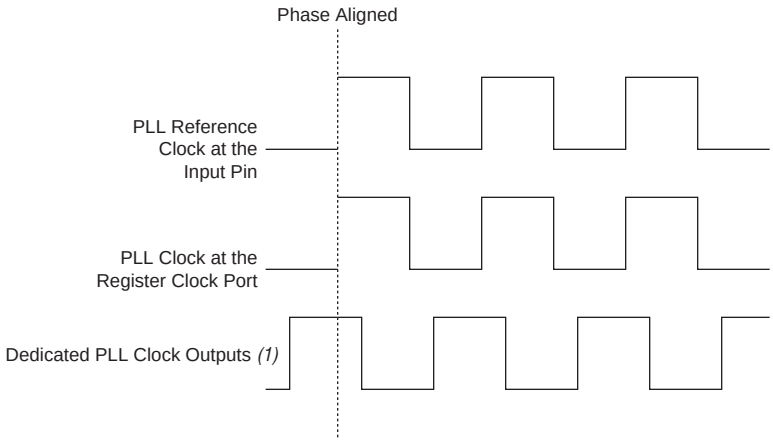
Note to Figure 4–27:

(1) The PLL clock outputs lag the PLL input clocks depending on routine delays.

Normal Mode

An internal clock in normal mode is phase-aligned to the input clock pin. The external clock-output pin has a phase delay relative to the clock input pin if connected in this mode. The Quartus II software timing analyzer reports any phase difference between the two. In normal mode, the delay introduced by the GCLK or RCLK network is fully compensated. Figure 4–28 shows an example waveform of the PLL clocks' phase relationship in normal mode.

Figure 4–28. Phase Relationship Between the PLL Clocks in Normal Mode



Note to Figure 4–28:

(1) The external clock output can lead or lag the PLL internal clock signals.

Zero-Delay Buffer Mode

In ZDB mode, the external clock output pin is phase-aligned with the clock input pin for zero delay through the device. When using this mode, you must use the same I/O standard on the input clocks and output clocks to guarantee clock alignment at the input and output pins. ZDB mode is supported only on the center and corner PLLs in Stratix V devices.

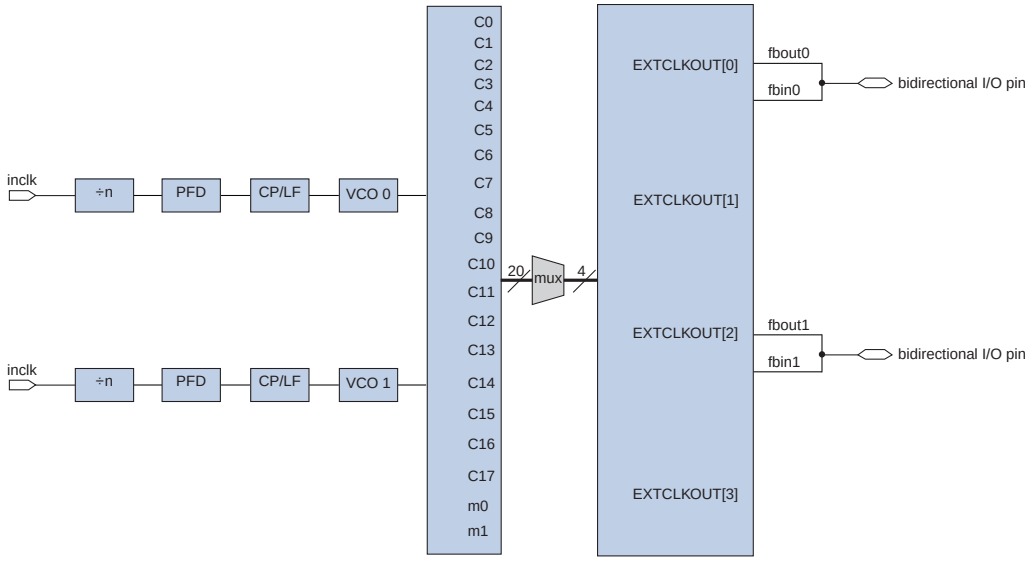
When using Stratix V PLLs in ZDB mode, along with single-ended I/O standards, to ensure phase alignment between the `CLK` pin and the external clock output (`CLKOUT`) pin, you must instantiate a bi-directional I/O pin in the design to serve as the feedback path connecting the `FBOUT` and `FBIN` ports of the PLL. The PLL uses this bidirectional I/O pin to mimic, and compensate for, the output delay from the clock output port of the PLL to the external clock output pin.

The bidirectional I/O pin that you instantiate in your design must always be assigned a single-ended I/O standard.

To avoid signal reflection when using ZDB mode, do not place board traces on the bi-directional I/O pin.

Figure 4–29 shows ZDB mode in Stratix V PLLs. When using ZDB mode, you cannot use differential I/O standards on the PLL clock input or output pins.

Figure 4–29. ZDB Mode in Stratix V PLLs *(1)*

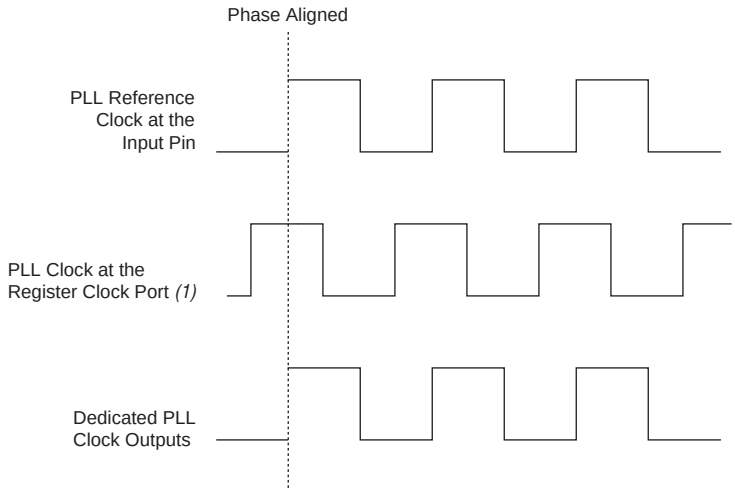


Note to Figure 4–29:

(1) ZDB mode can support up to four single-ended clock outputs. For more information, refer to Figure 4–24 on page 4–26.

Figure 4–30 shows an example waveform of the PLL clocks' phase relationship in ZDB mode.

Figure 4–30. Phase Relationship Between the PLL Clocks in ZDB Mode



Note to Figure 4–30:

(1) The internal PLL clock output can lead or lag the external PLL clock outputs.

External Feedback Mode

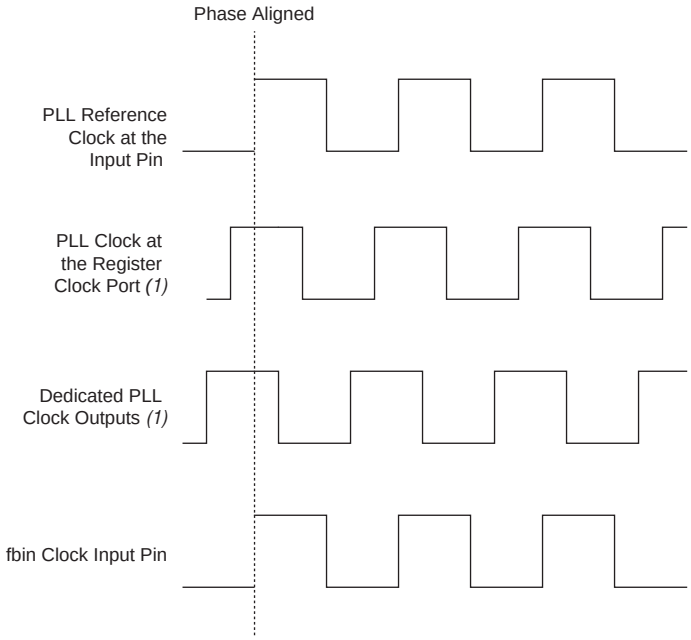
Figure 4–31 shows that in external feedback mode, the external feedback input pin (`fbin`) is phase-aligned with the clock input pin. Aligning these clocks allows you to remove clock delay and skew between devices. EFB mode is supported on only the center and corner PLLs in Stratix V devices.

In external feedback mode, the output of the `M` counter (`FBOUT`) feeds back to the PLL `fbin` input (using a trace on the board) and becomes part of the feedback loop. Also, one of the dual-purpose external clock outputs as the `fbin` input pin in this mode.

When using external feedback mode, you must use the same I/O standard on the input clock, feedback input, and output clocks. Left and right PLLs support this mode when using single-ended I/O standards only.

Figure 4–31 shows an example waveform of the phase relationship between the PLL clocks in external feedback mode.

Figure 4–31. Phase Relationship Between the PLL Clocks in External Feedback Mode

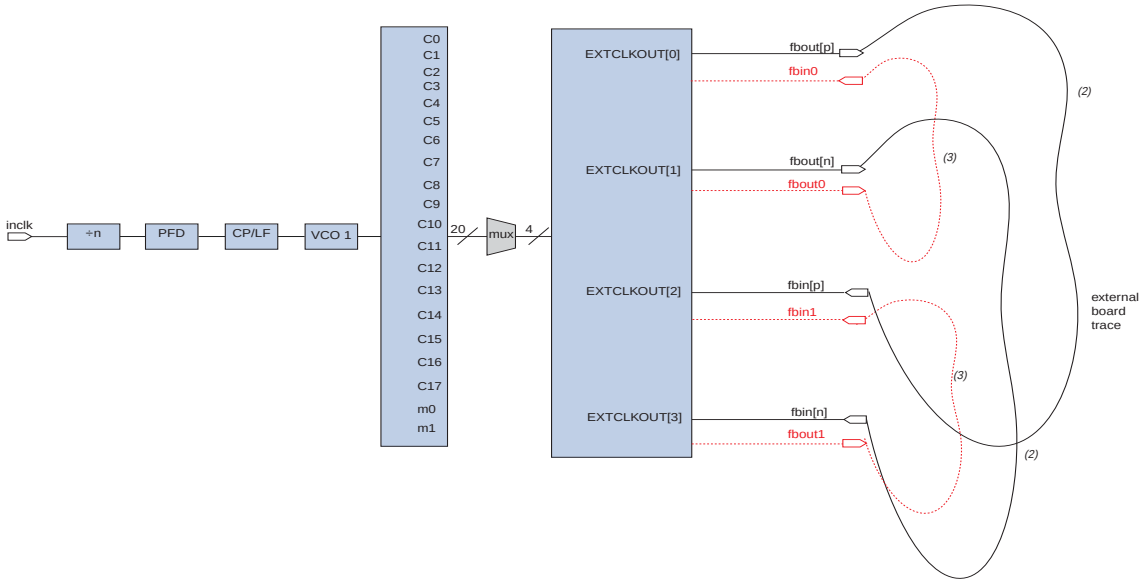

Phase Aligned
PLL Reference Clock at the Input Pin
PLL Clock at the Register Clock Port (1)
Dedicated PLL Clock Outputs (1)
fbin Clock Input Pin

Note to Figure 4–31:

(1) The PLL clock outputs can lead or lag the `fbin` clock input.

Figure 4–32 shows external feedback mode implementation in Stratix V devices.

Figure 4–32. External Feedback Mode in Stratix V Devices *(1)*



Notes to Figure 4–32:

(1) EFB mode can support two single-ended or one differential feedback inputs. For more information, refer to Figure 4–24 on page 4–26.
(2) External board connection for one differential clock output and one differential feedback input for differential EFB support.
(3) External board connection for two single-ended clock outputs and two single-ended feedback inputs for single-ended EFB support.

Implementing Multiple PLLs in Normal Mode and Source Synchronous Mode

Normal compensation mode and source synchronous compensation feedback mode require GCLK or RCLK feedback path to achieve the required phase relationship. Source synchronous mode for LVDS compensation does not require the GCLK or RCLK feedback path.

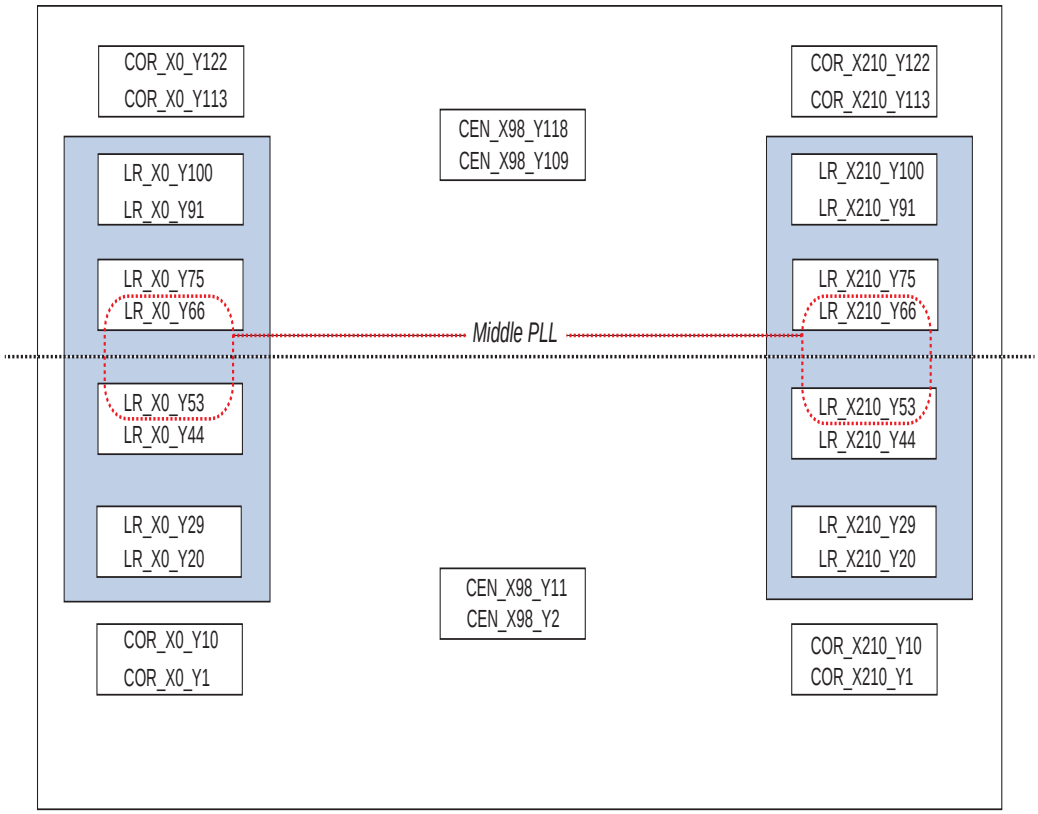
Stratix V devices have fewer GCLK or RCLK network feedback paths than the PLLs available on the device. You cannot implement the compensation mode that requires GCLK or RCLK feedback path on all the PLLs available on the device simultaneously.

Consider the following guidelines when implementing normal compensation or source synchronous compensation mode on multiple PLLs for the device:

- You can implement normal compensation or source synchronous compensation mode on all the center PLLs (CEN_X<#>_Y<#>) simultaneously.
- Stratix V device has two middle PLLs on the left and right side of the device. All PLLs that reside on each side of the device can be divided equally into 2 groups as shown in Figure 4–33.

Figure 4–33 shows an example of the PLL grouping for 5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 devices.

Figure 4–33. Example of the PLL Grouping for 5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 Devices



For the example shown in Figure 4–33, the PLLs can be divided into 4 different sections (upper left, lower left, upper right, and lower right). The PLLs in each of these sections can be further divided into first and second group. The first group consists of the 2 corner PLLs and one middle PLLs located in each section. The remaining PLLs in the same section are grouped into the second group. For each section, you can use up to 3 PLLs to implement source synchronous or normal compensation mode in the following combinations:

- Any of the 3 PLLs in the first group
- Any of the 2 PLLs in the first group and 1 PLL in the second group

Table 4–6 shows the PLL grouping for implementing source synchronous and normal compensation mode for 5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 devices.

Table 4–6. PLL Grouping for Implementing Source Synchronous and Normal Compensation Mode for 5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 Devices

PLL Section	PLL Location	
	First Group	Second Group
Upper Left	COR_X0_Y122, COR_X0_Y113, LR_X0_Y66	LR_X0_Y100, LR_X0_Y91, LR_X0_Y75
Lower Left	LR_X0_Y53, COR_X0_Y10, COR_X0_Y1	LR_X0_Y44, LR_X0_Y29, LR_X0_Y20
Upper Right	COR_X210_Y122, COR_X210_Y113, LR_X210_Y66	LR_X210_Y100, LR_X210_Y91, LR_X210_Y75
Lower Right	LR_X210_Y53, COR_X210_Y10, COR_X210_Y1	LR_X210_Y44, LR_X210_Y29, LR_X210_Y20

Clock Multiplication and Division

Each Stratix V PLL provides clock synthesis for PLL output ports using the $(K \times M)/(N \times C)$ scaling factors. The input clock is divided by a pre-scale factor, N, and is then multiplied by the K and M feedback factors. The control loop drives the VCO to match $f_{in} \times (K \times M/N)$.

A post-scale counter, K, is inserted after the VCO. When you enable the VCO post-scale counter, the counter divides the VCO frequency by two. When the K counter is bypassed, the VCO frequency goes to the output port without being divided by two.

Each output port has a unique post-scale counter, C, that divides down the output from the K counter. For multiple PLL outputs with different frequencies, the VCO is set to the least common multiple of the output frequencies that meets its frequency specifications. For example, if the output frequencies required from one PLL are 33 and 66 MHz, the Quartus II software sets the VCO to 660 MHz (the least common multiple of 33 and 66 MHz within the VCO range). Then the post-scale counters, C, scale down the VCO frequency for each output port.

Each PLL has one pre-scale counter, N, and one multiply counter, M, with a range of 1 to 512 for both M and N. For PLL operating in integer mode, M is an integer value. The delta-sigma modulator (DSM) is used together with M multiply counter to enable the PLL to operate in fractional mode. The DSM dynamically changes the M counter divide value on a cycle to cycle basis. The different M counter values lets the "average" M counter value to be a non-integer. In fractional mode, the M counter divide value equals to the sum of the "clock high" count, "clock low" count, and the fractional value. The fractional value is equal to K[31:0]/2^24. The PLL operates in integer mode when DSM is disabled.

The N counter does not use duty-cycle control because the only purpose of this counter is to calculate frequency division. The post-scale counters range from 1 to 512 with a 50% duty cycle setting. The high- and low-count values for each counter range from 1 to 256. The sum of the high- and low-count values chosen for a design selects the divide value for a given counter.

The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the ALTERA_PLL megafunction.

Programmable Duty Cycle

The programmable duty cycle allows PLLs to generate clock outputs with a variable duty cycle. This feature is supported on the PLL post-scale counters. The duty-cycle setting is achieved by a low and high time-count setting for the post-scale counters. To determine the duty cycle choices, the Quartus II software uses the frequency input and the required multiply or divide rate. The post-scale counter value determines the precision of the duty cycle. The precision is defined as 50% divided by the post-scale counter value. For example, if the `C0` counter is 10, steps of 5% are possible for duty-cycle choices from 5% to 90%.

If the PLL is in external feedback mode, set the duty cycle for the counter driving the `fbin` pin to 50%. Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

Clock Switchover

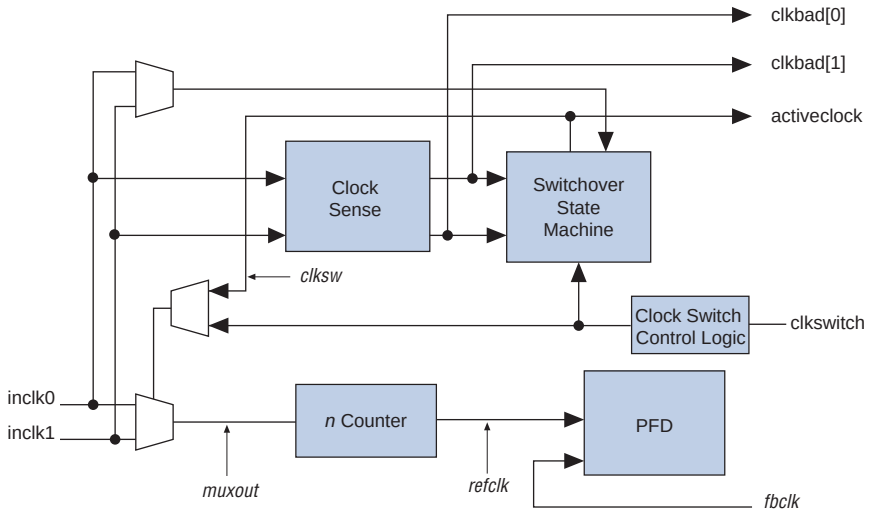
The clock switchover feature allows the PLL to switch between two reference input clocks. Use this feature for clock redundancy or for a dual-clock domain application such as in a system that turns on the redundant clock if the previous clock stops running. The design can perform clock switchover automatically when the clock is no longer toggling or based on a user control signal, `clkswitch`.

The following clock switchover modes are supported in Stratix V PLLs:

- Automatic switchover—The clock sense circuit monitors the current reference clock and if it stops toggling, automatically switches to the other `inclk0` or `inclk1` clock.
- Manual clock switchover—Clock switchover is controlled using the `clkswitch` signal. When the `clkswitch` signal goes from logic low to logic high, and stays high for at least three clock cycles, the reference clock to the PLL is switched from `inclk0` to `inclk1`, or vice-versa.
- Automatic switchover with manual override—This mode combines automatic switchover and manual clock switchover. When the `clkswitch` signal goes high, it overrides the automatic clock switchover function. As long as the `clkswitch` signal is high, further switchover action is blocked.

Stratix V PLLs support a fully configurable clock switchover capability. Figure 4–34 shows a block diagram of the automatic switchover circuit built into the PLL. When the current reference clock is not present, the clock sense block automatically switches to the backup clock for PLL reference. The clock switchover circuit also sends out three status signals—`clkbad[0]`, `clkbad[1]`, and `activeclock`—from the PLL to implement a custom switchover circuit in the logic array. You can select a clock source as the backup clock by connecting it to the `inclk1` port of the PLL in your design.

Figure 4–34. Automatic Clock Switchover Circuit Block Diagram



Automatic Clock Switchover

Use the switchover circuitry to automatically switch between `inclk0` and `inclk1` when the current reference clock to the PLL stops toggling. For example, in applications that require a redundant clock with the same frequency as the reference clock, the switchover state machine generates a signal (`clksw`) that controls the multiplexer select input, as shown in Figure 4–34. In this case, `inclk1` becomes the reference clock for the PLL. When using automatic switchover mode, you can switch back and forth between `inclk0` and `inclk1` any number of times when one of the two clocks fails and the other clock is available.

When using automatic clock switchover mode, the following requirements must be satisfied:

- Both clock inputs must be running
- The period of the two clock inputs can differ by no more than 20%

If the current clock input stops toggling while the other clock is also not toggling, switchover is not initiated and the `clkbad[0..1]` signals are not valid. Also, if both clock inputs are not the same frequency, but their period difference is within 20%, the clock sense block detects when a clock stops toggling, but the PLL may lose lock after the switchover is completed and needs time to re-lock.

Altera recommends resetting the PLL using the `areset` signal to maintain the phase relationships between the PLL input and output clocks when using clock switchover.

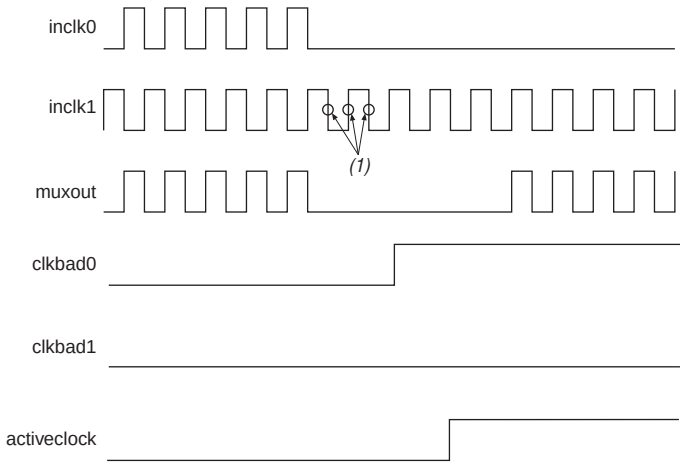
In automatic switchover mode, the clkbad[0] and clkbad[1] signals indicate the status of the two clock inputs. When they are asserted, the clock sense block has detected that the corresponding clock input has stopped toggling. These two signals are not valid if the frequency difference between inclk0 and inclk1 is greater than 20%.

The activeclock signal indicates which of the two clock inputs (inclk0 or inclk1) is being selected as the reference clock to the PLL. When the frequency difference between the two clock inputs is more than 20%, the activeclock signal is the only valid status signal.

Glitches in the input clock may cause the frequency difference between the input clocks to be more than 20%.

Figure 4–35 shows an example waveform of the switchover feature when using automatic switchover mode. In this example, the inclk0 signal is stuck low. After the inclk0 signal is stuck at low for approximately two clock cycles, the clock sense circuitry drives the clkbad[0] signal high. Also, because the reference clock signal is not toggling, the switchover state machine controls the multiplexer through the clkswitch signal to switch to the backup clock, inclk1.

Figure 4–35. Automatic Switchover After Loss of Clock Detection



Note to Figure 4–35:

(1) Switchover is enabled on the falling edge of inclk0 or inclk1, depending on which clock is available. In this figure, switchover is enabled on the falling edge of inclk1.

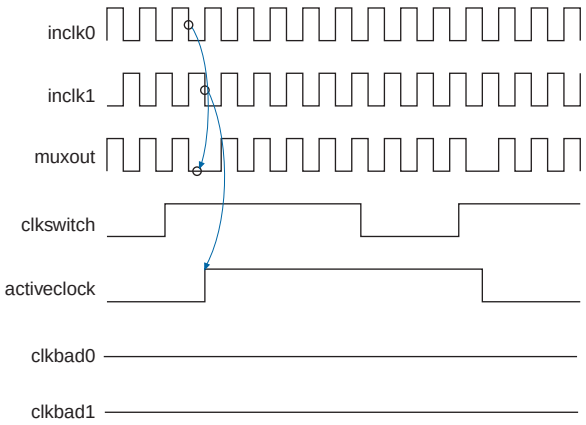
Manual Override

In automatic switchover with manual override mode, you can use the clkswitch input for user- or system-controlled switch conditions. You can use this mode for same-frequency switchover, or to switch between inputs of different frequencies. For example, if inclk0 is 66 MHz and inclk1 is 200 MHz, you must control switchover using clkswitch because the automatic clock-sense circuitry cannot monitor clock input (inclk0 and inclk1) frequencies with a frequency difference of more than 100% (2×). This feature is useful when the clock sources originate from multiple cards on the

backplane, requiring a system-controlled switchover between the frequencies of operation. You must choose the backup clock frequency and set the M, N, C, and K counters accordingly so that the VCO operates within the recommended operating frequency range. The ALTERA_PLL MegaWizard Plug-in Manager notifies you if a given combination of inclk0 and inclk1 frequencies cannot meet this requirement.

Figure 4–36 shows a clock switchover waveform controlled by clkswitch. In this case, both clock sources are functional and inclk0 is selected as the reference clock; clkswitch goes high, which starts the switchover sequence. On the falling edge of inclk0, the counter's reference clock, muxout, is gated off to prevent clock glitching. On the falling edge of inclk1, the reference clock multiplexer switches from inclk0 to inclk1 as the PLL reference and the activeclock signal changes to indicate which clock is currently feeding the PLL.

Figure 4–36. Clock Switchover Using the clkswitch (Manual) Control *(1)*



Note to Figure 4–36:

(1) To initiate a manual clock switchover event, both inclk0 and inclk1 must be running when the clkswitch signal goes high.

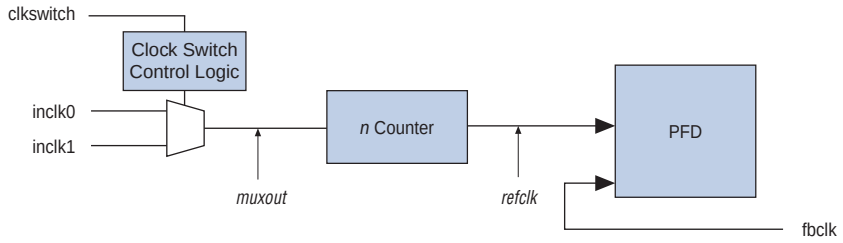
In automatic override with manual switchover mode, the activeclock signal mirrors the clkswitch signal. As both clocks are still functional during the manual switch, neither clkbad signal goes high. Because the switchover circuit is positive-edge sensitive, the falling edge of the clkswitch signal does not cause the circuit to switch back from inclk1 to inclk0. When the clkswitch signal goes high again, the process repeats. clkswitch and automatic switch only work if the clock being switched to is available. If the clock is not available, the state machine waits until the clock is available.

Manual Clock Switchover

In manual clock switchover mode, the `clkswitch` signal controls whether `inclk0` or `inclk1` is selected as the input clock to the PLL. By default, `inclk0` is selected. A low-to-high transition on `clkswitch` and `clkswitch` being held high for at least three `inclk` cycles initiates a clock switchover event.

You must bring `clkswitch` back low again to perform another switchover event in the future. If you do not require another switchover event in the future, you can leave `clkswitch` in a logic high state after the initial switch. Pulsing `clkswitch` high for at least three `inclk` cycles performs another switchover event. If `inclk0` and `inclk1` are different frequencies and are always running, the `clkswitch` minimum high time must be greater than or equal to three of the slower frequency `inclk0` or `inclk1` cycles. Figure 4–37 shows a block diagram of the manual switchover circuit.

Figure 4–37. Manual Clock Switchover Circuitry in Stratix V PLLs



You can delay the clock switchover action by specifying the switchover delay in the ALTERA_PLL megafunction. When you specify the switchover delay, the `clkswitch` signal must be held high for at least three `inclk` cycles plus the number of the delay cycles that has been specified to initiate a clock switchover.

For more information about PLL software support in the Quartus II software, refer to the *Altera Phase-Locked Loop (ALTERA_PLL) Megafunction User Guide*.

Guidelines

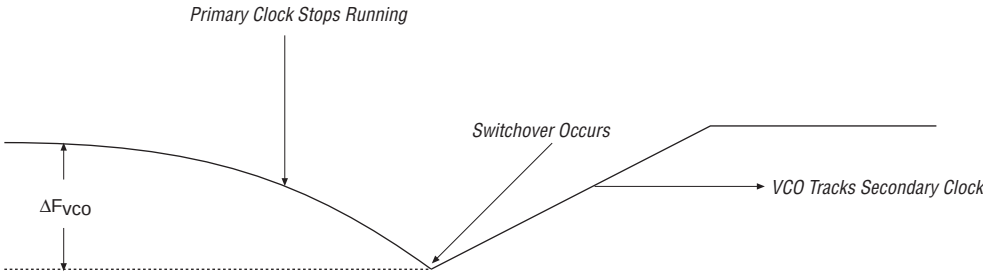
When implementing clock switchover in Stratix V PLLs, use the following guidelines:

- Automatic clock switchover requires that the `inclk0` and `inclk1` frequencies be within 20% of each other. Failing to meet this requirement causes the `clkbad[0]` and `clkbad[1]` signals to not function properly.
- When using manual clock switchover, the difference between `inclk0` and `inclk1` can be more than 100% (2×). However, differences in frequency, phase, or both, of the two clock sources will likely cause the PLL to lose lock. Resetting the PLL ensures that the correct phase relationships are maintained between the input and output clocks.

  Both `inclk0` and `inclk1` must be running when the `clkswitch` signal goes high to initiate the manual clock switchover event. Failing to meet this requirement causes the clock switchover to not function properly.

- Applications that require a clock switchover feature and a small frequency drift must use a low-bandwidth PLL. The low-bandwidth PLL reacts more slowly than a high-bandwidth PLL to reference input clock changes. When switchover happens, a low-bandwidth PLL propagates the stopping of the clock to the output more slowly than a high-bandwidth PLL. However, be aware that the low-bandwidth PLL also increases lock time.
- After a switchover occurs, there may be a finite resynchronization period for the PLL to lock onto a new clock. The exact amount of time it takes for the PLL to re-lock depends on the PLL configuration.
- The phase relationship between the input clock to the PLL and the output clock from the PLL is important in your design. Assert `areset` for at least 10 ns after performing a clock switchover. Wait for the locked signal to go high and be stable before re-enabling the output clocks from the PLL.
- Figure 4–38 shows how the VCO frequency gradually decreases when the current clock is lost and then increases as the VCO locks on to the backup clock.

Figure 4–38. VCO Switchover Operating Frequency



- Disable the system during clock switchover if it is not tolerant of frequency variations during the PLL resynchronization period. You can use the `clkbad[0]` and `clkbad[1]` status signals to turn off the PFD (`PFDENA = 0`) so the VCO maintains its most recent frequency. You can also use the state machine to switch over to the secondary clock. When the PFD is re-enabled, output clock-enable signals (`clkena`) can disable clock outputs during the switchover and resynchronization period. When the lock indication is stable, the system can re-enable the output clocks.

PLL Reconfiguration and Dynamic Phase Shift

The PLLs in Stratix V devices support dynamic reconfiguration and dynamic phase shift. While the device is in user mode, you can download a new PLL configuration or adjust the phase shift of the PLL output clock in real time without reconfiguring the entire FPGA.

For more information about PLL reconfiguration and dynamic phase shift, refer to *AN661: Implementing Fractional PLL Reconfiguration with ALTERA_PLL and ALTERA_PLL_RECONFIG Megafunctions*.

Document Revision History

Table 4–7 lists the revision history for this chapter.

Table 4–7. Document Revision History

Date	Version	Changes
June 2012	1.4	■ Added Table 4–5 and Table 4–6. ■ Added Figure 4–6, Figure 4–8, Figure 4–20, Figure 4–22, and Figure 4–33. ■ Updated Table 4–1, Table 4–2, and Table 4–3. ■ Updated Figure 4–3, Figure 4–5, Figure 4–17, Figure 4–18, Figure 4–19, and Figure 4–21. ■ Added "PLL Migration Guidelines", "Implementing Multiple PLLs in Normal Mode and Source Synchronous Mode", "Clock Switchover", and "PLL Reconfiguration and Dynamic Phase Shift" sections. ■ Updated "Clock Networks in Stratix V Devices", "Clock Network Sources", and "Clock Multiplication and Division" sections.
November 2011	1.3	Updated Figure 4–19 and Figure 4–28.
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 4–1. ■ Updated Figure 4–3, Figure 4–4, Figure 4–5, Figure 4–6, Figure 4–15, Figure 4–17, Figure 4–18, Figure 4–20, Figure 4–25, and Figure 4–28. ■ Updated "Zero-Delay Buffer Mode" and "External Feedback Mode" sections. ■ Added "PLL Clock Outputs" section.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

Section II. I/O Interfaces

This section provides information about Stratix® V device I/O features, external memory interfaces, and high-speed differential interfaces with dynamic phase alignment (DPA). This section includes the following chapters:

- Chapter 5, I/O Features in Stratix V Devices
- Chapter 6, High-Speed Differential I/O Interfaces and DPA in Stratix V Devices
- Chapter 7, External Memory Interfaces in Stratix V Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

5. I/O Features in Stratix V Devices

SV51006-1.5

This chapter describes how Stratix® V devices provide I/O capabilities that allow you to work in compliance with current and emerging I/O standards and requirements. With these device features, you can reduce board design interface costs and increase development flexibility.

Stratix V I/Os are specifically designed for ease-of-use and rapid system integration, while simultaneously providing the high bandwidth required to maximize internal logic capabilities and enhance system-level performance.

The I/O capability of the Stratix V device family exceeds the I/O bandwidth available in previous generation FPGAs. Independent modular I/O banks with a common bank structure for vertical migration lend efficiency and flexibility to the high-speed I/Os.

Package and die enhancements with dynamic termination and output control provide the best signal integrity in its class. Stratix V devices provide I/O features that assist high-speed data transfer into and out of the device, including the following features:

- Up to 1020 general purpose I/Os (GPIOs) and 255 full-duplex true LVDS channels
- True LVDS channels in all I/O banks support SGMII, SPI-4.2, and XSBI applications
- Hard dynamic phase alignment (DPA) and serializer/deserializer (SERDES) support in I/O banks on all sides of the device with DPA
- Single-ended, non-voltage-referenced, and voltage-referenced I/O standards
- **LVDS**, **RSDS**, **mini-LVDS**, **HSTL**, **SSTL**, and **HSUL** I/O standards across all I/O banks
- Double data rate (DDR), single data rate (SDR), and half data rate input and output options
- Ubiquitous I/O support for both row and column I/Os
- Deskew, read and write leveling, and clock-domain crossing functionality for high-performance memory interface
- Programmable output current strength
- Programmable slew rate
- Programmable input and output delays
- Programmable bus-hold circuits
- Programmable pull-up resistors
- Open-drain output

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

- Dynamic on-chip termination (OCT) for on-chip series (R_S) with and without calibration, and on-chip parallel (R_T) termination with calibration
- Differential (R_D) OCT without calibration
- Programmable pre-emphasis
- Programmable differential output voltage (V_{OD})

The following information is applicable to all Stratix V variants, unless noted otherwise.

I/O Standard Support

Stratix V devices support a wide range of industry I/O standards. Table 5–1 lists the I/O standards for Stratix V devices, as well as the typical applications they support. These devices support V_{CCIO} voltage levels of 3.0, 2.5, 1.8, 1.5, 1.35, 1.25, and 1.2 V.

Table 5–1. Stratix V I/O Standards and Applications (Part 1 of 2)

I/O Standard	Typical Applications
3.3-V LVTTL/LVCMOS *(1), (2)*	General purpose
2.5-V LVCMOS	General purpose
1.8-V LVCMOS	General purpose
1.5-V LVCMOS	General purpose
1.2-V LVCMOS	General purpose
SSTL-2 Class I and II	DDR SDRAM
SSTL-18 Class I and II	DDR2 SDRAM
SSTL-15 Class I and II	DDR3 SDRAM
SSTL-15	DDR3 SDRAM
SSTL-135	DDR3L SDRAM
SSTL-125	DDR3U SDRAM
SSTL-12	RLDRAM III
HSTL-18 Class I and II	QDR II/RLDRAM II
HSTL-15 Class I and II	QDR II/QDR II+/RLDRAM II
HSTL-12 Class I and II	General purpose
HSUL-12	LPDDR2 SDRAM
Differential SSTL-2 Class I and II	DDR SDRAM
Differential SSTL-18 Class I and II	DDR2 SDRAM
Differential SSTL-15 Class I and II	DDR3 SDRAM
Differential HSTL-18 Class I and II	Clock interfaces
Differential HSTL-15 Class I and II	Clock interfaces
Differential HSTL-12 Class I and II	Clock interfaces
Differential SSTL-15	DDR3 SDRAM
Differential SSTL-135	DDR3L SDRAM
Differential SSTL-125	DDR3U SDRAM
Differential SSTL-12	RLDRAM III

Table 5–1. Stratix V I/O Standards and Applications (Part 2 of 2)

I/O Standard	Typical Applications
Differential HSUL-12	LPDDR2 SDRAM
LVDS	High-speed communications
RSDS	Flat panel display
mini-LVDS	Flat panel display
LVPECL	Video graphics and clock distribution

Notes to Table 5–1:

(1) The **3.3-V LVTTL/LVCMOS** I/O standard is supported using V_{CCIO} at 3.0 V.

(2) This I/O standard is only supported in Stratix V GX and GS devices.

I/O Standards and Voltage Levels

Table 5–2 lists the supported I/O standards and typical V_{CCIO}, V_{CCPD}, V_{REF}, and board V_{TT} values for input and output.

Table 5–2. Stratix V I/O Standards and Voltage Levels (1) (Part 1 of 2)

I/O Standard	Standard Support	V_{CCIO} (V)		V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
		Input Operation	Output Operation			
3.3-V LVTTL (2)	JESD8-B	3.0/2.5	3.0	3.0	—	—
3.3-V LVCMOS (2)	JESD8-B	3.0/2.5	3.0	3.0	—	—
2.5-V LVCMOS	JESD8-5	3.0/2.5	2.5	2.5	—	—
1.8-V LVCMOS	JESD8-7	1.8/1.5	1.8	2.5	—	—
1.5-V LVCMOS	JESD8-11	1.8/1.5	1.5	2.5	—	—
1.2-V LVCMOS	JESD8-12	1.2	1.2	2.5	—	—
SSTL-2 Class I	JESD8-9B	(3)	2.5	2.5	1.25	1.25
SSTL-2 Class II	JESD8-9B	(3)	2.5	2.5	1.25	1.25
SSTL-18 Class I	JESD8-15	(3)	1.8	2.5	0.90	0.90
SSTL-18 Class II	JESD8-15	(3)	1.8	2.5	0.90	0.90
SSTL-15 Class I	—	(3)	1.5	2.5	0.75	0.75
SSTL-15 Class II	—	(3)	1.5	2.5	0.75	0.75
SSTL-15	JESD79-3D	(3)	1.5	2.5	0.75	(4)
SSTL-135	—	(3)	1.35	2.5	0.675	(4)
SSTL-125	—	(3)	1.25	2.5	0.625	(4)
SSTL-12	—	(3)	1.2	2.5	0.6	(4)
HSTL-18 Class I	JESD8-6	(3)	1.8	2.5	0.90	0.90
HSTL-18 Class II	JESD8-6	(3)	1.8	2.5	0.90	0.90
HSTL-15 Class I	JESD8-6	(3)	1.5	2.5	0.75	0.75
HSTL-15 Class II	JESD8-6	(3)	1.5	2.5	0.75	0.75
HSTL-12 Class I	JESD8-16A	(3)	1.2	2.5	0.6	0.6
HSTL-12 Class II	JESD8-16A	(3)	1.2	2.5	0.6	0.6
HSUL-12	—	(3)	1.2	2.5	0.6	(4)

Table 5–2. Stratix V I/O Standards and Voltage Levels (1) (Part 2 of 2)

I/O Standard	Standard Support	V_{CCIO} (V)		V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
		Input Operation	Output Operation			
Differential SSTL-2 Class I	JESD8-9B	(3)	2.5	2.5	—	1.25
Differential SSTL-2 Class II	JESD8-9B	(3)	2.5	2.5	—	1.25
Differential SSTL-18 Class I	JESD8-15	(3)	1.8	2.5	—	0.90
Differential SSTL-18 Class II	JESD8-15	(3)	1.8	2.5	—	0.90
Differential SSTL-15 Class I	—	(3)	1.5	2.5	—	0.75
Differential SSTL-15 Class II	—	(3)	1.5	2.5	—	0.75
Differential HSTL-18 Class I	JESD8-6	(3)	1.8	2.5	—	0.90
Differential HSTL-18 Class II	JESD8-6	(3)	1.8	2.5	—	0.90
Differential HSTL-15 Class I	JESD8-6	(3)	1.5	2.5	—	0.75
Differential HSTL-15 Class II	JESD8-6	(3)	1.5	2.5	—	0.75
Differential HSTL-12 Class I	JESD8-16A	(3)	1.2	2.5	—	0.60
Differential HSTL-12 Class II	JESD8-16A	(3)	1.2	2.5	—	0.60
Differential SSTL-15	JESD79-3D	(3)	1.5	2.5	—	(4)
Differential SSTL-135	—	(3)	1.35	2.5	—	(4)
Differential SSTL-125	—	(3)	1.25	2.5	—	(4)
Differential SSTL-12	—	(3)	1.2	2.5	—	(4)
Differential HSUL-12	—	(3)	1.2	2.5	—	(4)
LVDS (6), (7)	ANSI/TIA/EIA-644	(3)	2.5	2.5	—	—
RSDS (6), (7)	—	(3)	2.5	2.5	—	—
mini-LVDS (6), (7)	—	(3)	2.5	2.5	—	—
LVPECL	—	(5)	—	2.5	—	—

Notes to Table 5–2:

(1) V_{CCPD} is either 2.5 or 3.0 V. For V_{CCIO} = 3.0 V, V_{CCPD} = 3.0 V. For V_{CCIO} = 2.5 V or less, V_{CCPD} = 2.5 V.

(2) For more information about the **3.3-V LVTTL/LVCMOS** I/O standard supported in Stratix V devices, refer to "3.3-V I/O Interface" on page 5–9.

(3) Single-ended **HSTL**/**SSTL**/**HSUL**, **differential SSTL**/**HSTL**/**HSUL**, and **LVDS** input buffers are powered by V_{CCPD}. **Differential HSTL**, **SSTL**, and **HSUL** outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted. **Differential HSTL**, **SSTL**, and **HSUL** inputs use **LVDS** differential input buffers with R_D support.

(4) Typically this I/O standard does not require board termination.

(5) The **LVPECL** I/O standard is supported for input clock operation. Differential clock input buffers are powered by V_{CCPD}. When clock pins are configured as a single-ended I/O standard, these pins are powered by V_{CCIO}.

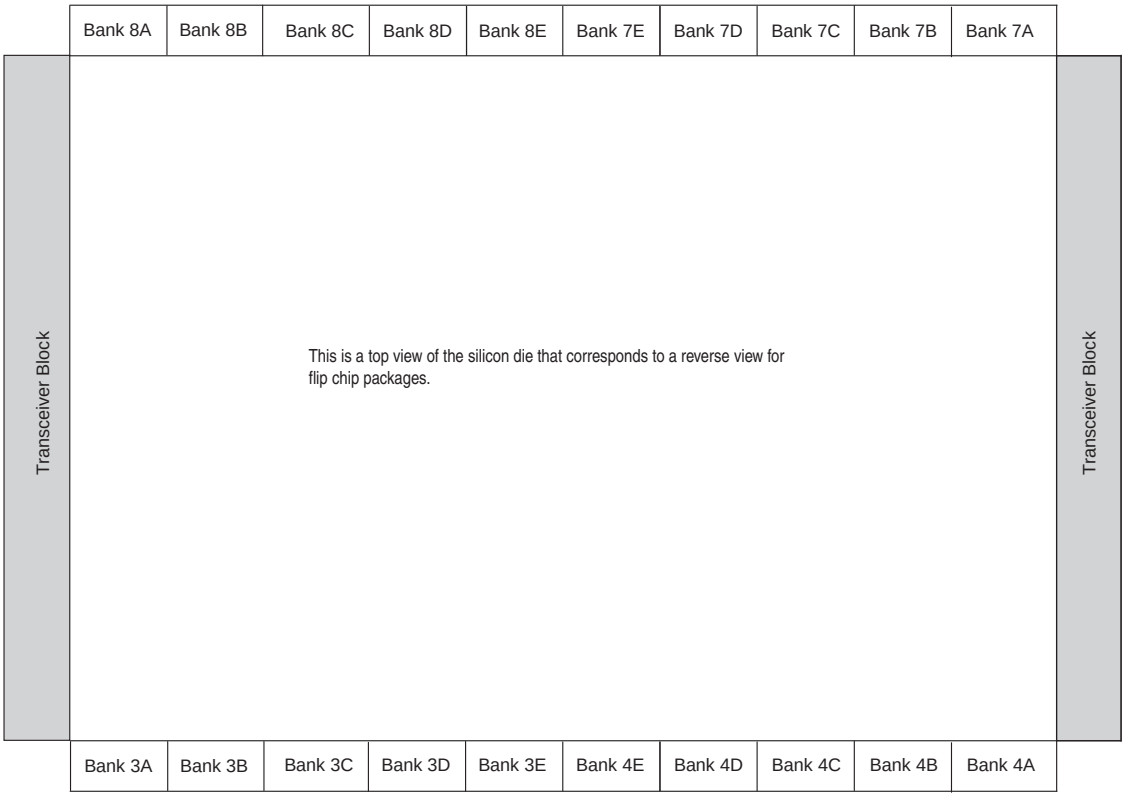
(6) All I/O banks support true **LVDS**, **RSDS**, and **mini-LVDS** I/O standards using true **LVDS** output buffers without resistor networks. All I/O banks also support emulated **LVDS**, **RSDS**, and **mini-LVDS** I/O standards using two single-ended output buffers with a three-resistor (**LVDS_E_3R**, **RSDS_E_3R**, and **mini-LVDS_E_3R**) network.

(7) The emulated differential output standard that supports the tri-state feature includes: **LVDS_E_3R**, **RSDS_E_3R**, and **mini_LVDS_E_3R**.

I/O Banks

Figure 5–1 shows the I/O banks in Stratix V devices. All I/O banks in Stratix V devices contain true differential input and output buffers and dedicated circuitry to support differential I/O standards. Each I/O bank in Stratix V devices supports a high-performance external memory interface. The I/O pins are organized in pairs to support differential I/O standards. Each I/O pin pair can support both differential input and output buffers.

Figure 5–1. I/O Banks for Stratix V Devices—Preliminary



Modular I/O Banks

The I/O pins in Stratix V devices are arranged in groups called modular I/O banks. The number of Stratix V I/O banks in a particular device ranges from 11 to 20, depending on the device density.

Table 5–3 through Table 5–5 list the modular I/O banks for Stratix V devices.

Table 5–3. Modular I/O Banks for Stratix V GX Devices—Preliminary

Package	Device	Bank																			Total
		3A	3B	3C	3D	3E	4A	4B	4C	4D	4E	7A	7B	7C	7D	7E	8A	8B	8C	8D	8E	
EH29	5SGXA3	36	48	—	24	—	24	—	—	24	—	24	—	48	36	—	24	—	48	24	—	360
HF35	5SGXA3	36	48	—	24	—	24	48	—	36	—	24	48	48	36	—	24	—	—	36	—	432
	5SGXA4	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
	5SGXA5	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
	5SGXA7	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
KF35	5SGXA3	36	48	—	24	—	24	48	—	36	—	24	48	48	36	—	24	—	—	36	—	432
	5SGXA4	36	48	—	24	—	24	48	—	36	—	24	48	48	36	—	24	—	—	36	—	432
	5SGXA5	36	48	—	24	—	24	48	—	36	—	24	48	48	36	—	24	—	—	36	—	432
	5SGXA7	36	48	—	24	—	24	48	—	36	—	24	48	48	36	—	24	—	—	36	—	432
KF40	5SGXA3	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGXA4	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGXA5	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGXA7	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
KH40	5SGXA9	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGXAB	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
NF40	5SGXA5	36	48	48	24	—	24	48	48	24	—	24	48	48	48	—	36	—	48	48	—	600
	5SGXA7	36	48	48	24	—	24	48	48	24	—	24	48	48	48	—	36	—	48	48	—	600
RF40	5SGXB5	36	48	—	—	—	48	48	36	—	—	48	48	36	—	—	36	48	—	—	—	432
	5SGXB6	36	48	—	—	—	48	48	36	—	—	48	48	36	—	—	36	48	—	—	—	432
RF43	5SGXB5	36	48	48	36	—	48	48	36	—	—	48	48	36	—	—	36	48	48	36	—	600
	5SGXB6	36	48	48	36	—	48	48	36	—	—	48	48	36	—	—	36	48	48	36	—	600
RH43	5SGXB9	36	48	48	36	—	48	48	36	—	—	48	48	36	—	—	36	48	48	36	—	600
	5SGXBB	36	48	48	36	—	48	48	36	—	—	48	48	36	—	—	36	48	48	36	—	600
NF45	5SGXA5	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SGXA7	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SGXA9	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SGXAB	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840

Table 5–4. Modular I/O Banks for Stratix V GS Devices—Preliminary

Package	Device	Bank																				Total
		3A	3B	3C	3D	3E	4A	4B	4C	4D	4E	7A	7B	7C	7D	7E	8A	8B	8C	8D	8E	
EF29	5SGSD3	36	48	—	24	—	24	—	—	24	—	24	—	48	36	—	24	—	48	24	—	360
	5SGSD4	36	48	—	24	—	24	—	—	24	—	24	—	48	36	—	24	—	48	24	—	360
HF35	5SGSD3	36	48	—	24	—	24	48	—	24	—	24	24	48	36	—	24	—	48	24	—	432
	5SGSD4	36	48	—	24	—	24	48	—	24	—	24	24	48	36	—	24	—	48	24	—	432
	5SGSD5	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
KF40	5SGSD4	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGSD5	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGSD6	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGSD8	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
NF45	5SGSD6	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SGSD8	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840

Table 5–5. Modular I/O Banks for Stratix V GT and E Devices—Preliminary

Package	Device	Bank																				Total
		3A	3B	3C	3D	3E	4A	4B	4C	4D	4E	7A	7B	7C	7D	7E	8A	8B	8C	8D	8E	
KF40	5SGTC5	36	48	48	24	—	24	48	48	24	—	24	48	48	48	—	36	—	48	48	—	600
	5SGTC7	36	48	48	24	—	24	48	48	24	—	24	48	48	48	—	36	—	48	48	—	600
H40	5SEE9	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SEEB	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
F45	5SEE9	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SEEB	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840

I/O Structure

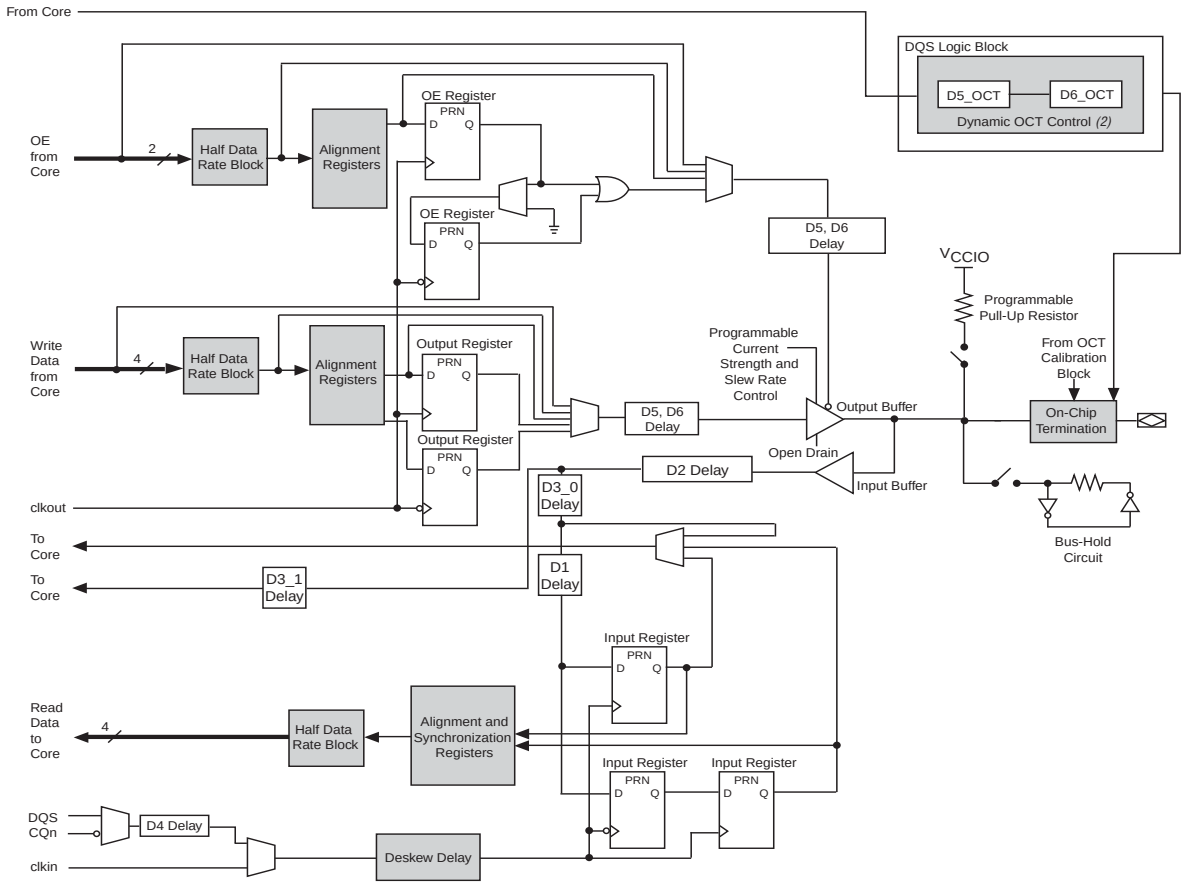
The I/O elements (IOEs) in Stratix V devices contain a bidirectional I/O buffer and I/O registers to support a complete embedded bidirectional SDR or DDR transfer. The IOEs are located in I/O blocks around the periphery of the Stratix V device.

I/O registers are composed of the input path for handling data from the pin to the core, the output path for handling data from the core to the pin, and the output enable, OE path for handling the OE signal to the output buffer. These registers allow faster source-synchronous register-to-register transfers and resynchronization. The input path consists of the DDR input registers, alignment and synchronization registers, and half data rate blocks; you can bypass each block in the input path. The input path uses the deskew delay to adjust the input register clock delay across process, voltage, and temperature (PVT) variations.

The output and OE paths are divided into the output or OE registers, alignment registers, and half data rate blocks. You can bypass each block of the output and OE paths.

Figure 5–2 shows the Stratix V IOE structure.

Figure 5–2. IOE Structure for Stratix V Devices *(1), (2)*



Notes to Figure 5–2:

(1) The D3_0 and D3_1 delays have the same available settings in the Quartus® II software.

(2) One dynamic OCT control is available per DQ/DQS group.

3.3-V I/O Interface

Stratix V I/O buffers support 3.3-V I/O standards. You can use them as transmitters or receivers in your system. The output high voltage (V_{OH}), output low voltage (V_{OL}), input high voltage (V_{IH}), and input low voltage (V_{IL}) levels meet the 3.3-V I/O standards specifications defined by EIA/JEDEC Standard JESD8-B with margin when the Stratix V V_{CCIO} voltage is powered by 3.0 V.

To ensure device reliability and proper operation when interfacing with a 3.3-V I/O system using Stratix V devices, do not violate the absolute maximum ratings of the devices. Altera recommends performing IBIS or SPICE simulations to determine that the overshoot and undershoot voltages are within the specifications.

When using a Stratix V device as a transmitter, you can use slow slew rate and series termination to limit overshoot and undershoot at the I/O pins. Transmission line effects that cause large voltage deviations at the receiver are associated with an impedance mismatch between the driver and the transmission lines. By matching the impedance of the driver to the characteristic impedance of the transmission line, you can significantly reduce overshoot voltage. You can use a series termination resistor placed physically close to the driver to match the total driver impedance to the transmission line impedance. Stratix V devices support R_S OCT for all **LVTTL** and **LVCMOS** I/O standards in all I/O banks.

When using the Stratix V device as a receiver, you can use a clamping diode (off-chip) if it is required to limit the overshoot voltage.

The 3.3-V I/O standard is supported using the bank supply voltage (V_{CCIO}) at 3.0 V and a V_{CCPD} voltage of 3.0 V. In this method, the clamping diode (off-chip) can sufficiently clamp overshoot voltage to within the DC and AC input voltage specifications. The clamped voltage is expressed as the sum of the V_{CCIO} and the diode forward voltage.

External Memory Interfaces

In addition to the I/O registers in each IOE, Stratix V devices also have dedicated registers and phase-shift circuitry on all I/O banks to interface with external memory. Stratix V devices support new I/O standards such as **SSTL-12**, **SSTL-15**, **SSTL-125**, **SSTL-135**, and **HSUL-12**.

High-Speed Differential I/O with DPA Support

Stratix V devices have the following dedicated circuitry for high-speed differential I/O support:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer
- Data realignment
- DPA
- Synchronizer (FIFO buffer)
- Phase-locked loops (PLLs)

Current Strength

The output buffer for each Stratix V device I/O pin has programmable current strength control for certain I/O standards. Use programmable current strength to mitigate the effects of high signal attenuation due to a long transmission line or a legacy backplane. The **LVTTL**, **LVCMOS**, **SSTL**, and **HSTL** I/O standards have several levels of current strength that you can control. Table 5–6 lists the programmable current strength settings for Stratix V devices.

Table 5–6. Programmable Current Strength Settings

I/O Standard	I_{OH} / I_{OL} Current Strength Setting	Units
3.3-V LVTTL (1)	16, 12, 8, 4	mA
3.3-V LVCMOS (1)	16, 12, 8, 4	mA
2.5-V LVCMOS	16, 12, 8, 4	mA
1.8-V LVCMOS	12, 10, 8, 6, 4, 2	mA
1.5-V LVCMOS	12, 10, 8, 6, 4, 2	mA
1.2-V LVCMOS	8, 6, 4, 2	mA
SSTL-2 Class I	12, 10, 8	mA
SSTL-2 Class II	16	mA
SSTL-18 Class I	12, 10, 8, 6, 4	mA
SSTL-18 Class II	16	mA
SSTL-15 Class I	12, 10, 8, 6, 4	mA
SSTL-15 Class II	16	mA
HSTL-18 Class I	12, 10, 8, 6, 4	mA
HSTL-18 Class II	16	mA
HSTL-15 Class I	12, 10, 8, 6, 4	mA
HSTL-15 Class II	16	mA
HSTL-12 Class I	12, 10, 8, 6, 4	mA
HSTL-12 Class II	**16**	mA

Note to Table 5–6:

(1) The **3.3-V LVTTL** and **3.3-V LVCMOS** I/O standards are supported using V_{CCIO} and V_{CCPD} at 3.0 V.

Altera recommends performing IBIS or SPICE simulations to determine the best current strength setting for your specific application.

Slew-Rate Control

The output buffer for each Stratix V device regular- and dual-function I/O pin has a programmable output slew rate control that you can configure for low-noise or high-speed performance. A fast slew rate provides high-speed transitions for high-performance systems. A slow slew rate can help reduce system noise, but adds a nominal delay to the rising and falling edges. Each I/O pin has an individual slew rate control, allowing you to specify the slew rate on a pin-by-pin basis.

☞ You cannot use the programmable slew rate feature when using R_S OCT.

The Quartus II software allows two settings for programmable slew rate control—0 and 1—where 0 is slow slew rate and 1 (default) is fast slew rate. Fast slew rates improve the available timing margin in memory-interface applications or when the output pin has high-capacitive loading.

☞ Altera recommends performing IBIS or SPICE simulations to determine the best slew rate setting for your specific application.

I/O Delay

The following sections describe programmable IOE delay and programmable output buffer delay.

Programmable IOE Delay

The Stratix V device IOE includes programmable delays, shown in Figure 5–2 on page 5–8 that you can activate to ensure zero hold times, minimize setup times, or increase clock-to-output times. Each single-ended and differential I/O pin can have a different input delay from pin-to-input register or a delay from output register-to-output pin values to ensure that the bus has the same delay going into or out of the device. This feature helps read and write timing margins because it minimizes the uncertainties between signals in the bus.

For more information about programmable IOE delay specifications, refer to the *Stratix V Device Datasheet*.

Programmable Output Buffer Delay

Stratix V devices support delay chains built inside the single-ended output buffer, as shown in Figure 5–2 on page 5–8. The delay chains can independently control the rising and falling edge delays of the output buffer, providing the ability to adjust the output-buffer duty cycle, compensate channel-to-channel skew, reduce simultaneous switching output (SSO) noise by deliberately introducing channel-to-channel skew, and improve high-speed memory-interface timing margins. Stratix V devices support four levels of output buffer delay settings with the default setting of no delay.

For more information about programmable output buffer delay specifications, refer to the *Stratix V Device Datasheet*.

Open-Drain Output

Stratix V devices provide an optional open-drain output (equivalent to an open collector output) for each I/O pin. When configured as open drain, the logic value of the output is either high-Z or 0. Typically, an external pull-up resistor is required to provide logic high.

Bus-Hold

Each Stratix V device I/O pin provides an optional bus-hold feature. Bus-hold circuitry can weakly hold the signal on an I/O pin at its last-driven state. Because the bus-hold feature holds the last-driven state of the pin until the next input signal is present, you do not require an external pull-up or pull-down resistor to hold a signal level when the bus is tri-stated.

Bus-hold circuitry also pulls non-driven pins away from the input threshold voltage where noise can cause unintended high-frequency switching. You can select this feature individually for each I/O pin. The bus-hold output drives no higher than the V_{CCIO} to prevent over-driving signals. If you enable the bus-hold feature, you cannot use the programmable pull-up option. Disable the bus-hold feature if the I/O pin is configured for differential signals.

Bus-hold circuitry uses a resistor with a nominal resistance (R_{BH}) of approximately 7 kΩ to weakly pull the signal level to the last-driven state.

The bus-hold circuit is active only after configuration. When going into user mode, the bus-hold circuit captures the value on the pin that is present at the end of configuration.

Pull-Up Resistor

Each Stratix V device I/O pin provides an optional programmable pull-up resistor during user mode. If you enable this feature for an I/O pin, the pull-up resistor (typically 25 kΩ) weakly holds the I/O to the V_{CCIO} level.

Programmable pull-up resistors are only supported on user I/O pins and are not supported on dedicated configuration pins or JTAG pins. If you enable the programmable pull-up option, you cannot use the bus-hold feature.

Pre-Emphasis

Stratix V LVDS transmitters support programmable pre-emphasis to compensate for the frequency dependent attenuation of the transmission line. The Quartus II software allows two settings for programmable pre-emphasis control—0 and 1—where 0 is disabled and 1 (default) is enabled.

Differential Output Voltage

Stratix V LVDS transmitters support programmable V_{OD}. The programmable V_{OD} settings allow you to adjust output eye height to optimize trace length and power consumption. A higher V_{OD} swing improves voltage margins at the receiver end; a smaller V_{OD} swing reduces power consumption. The Quartus II software allows four settings for programmable V_{OD} control—0, 1, 2, and 3—where 0 is low, 1 (default) is medium low, 2 is medium high, and 3 is high. The low setting for the programmable V_{OD} control is available for the **RSDS** I/O standard only and is not available for the **LVDS** I/O standard.

MultiVolt I/O Interface

The Stratix V architecture supports the MultiVolt I/O interface feature that allows Stratix V devices in all packages to interface with systems of different supply voltages.

You can connect the VCCIO pins to a 1.2-, 1.25-, 1.35-, 1.5-, 1.8-, 2.5-, or 3.0-V power supply, depending on the output requirements. The output levels are compatible with systems of the same voltage as the power supply. (For example, when VCCIO pins are connected to a 1.5-V power supply, the output levels are compatible with 1.5-V systems.)

The Stratix V VCCPD power pins must be connected to a 2.5- or 3.0-V power supply. Using these power pins to supply the pre-driver power to the output buffers increases the performance of the output pins. Table 5–7 lists Stratix V MultiVolt I/O support.

Table 5–7. Stratix V MultiVolt I/O Support *(1)*

V_{CCIO} (V) *(3)*	Input Signal (V)								Output Signal (V)							
	1.2	1.25	1.35	1.5	1.8	2.5	3.0	3.3	1.2	1.25	1.35	1.5	1.8	2.5	3.0	3.3
1.2	Y	—	—	—	—	—	—	—	Y	—	—	—	—	—	—	—
1.25	—	Y	—	—	—	—	—	—	—	Y	—	—	—	—	—	—
1.35	—	—	Y	—	—	—	—	—	—	—	Y	—	—	—	—	—
1.5	—	—	—	Y	Y	—	—	—	—	—	—	Y	—	—	—	—
1.8	—	—	—	Y	Y	—	—	—	—	—	—	—	Y	—	—	—
2.5	—	—	—	—	—	Y	Y *(2)*	Y *(2)*	—	—	—	—	—	Y	—	—
3.0	—	—	—	—	—	Y	Y *(2)*	Y *(2)*	—	—	—	—	—	—	Y	Y

Notes to Table 5–7:

(1) The pin current may be slightly higher than the default value. You must verify that the driving device's V_{OL} maximum and V_{OH} minimum voltages do not violate the applicable Stratix V V_{IL} maximum and V_{IH} minimum voltage specifications.

(2) Altera recommends that you use an external clamping diode on the I/O pins when the input signal is 3.0 V or 3.3 V.

(3) Each I/O bank of a Stratix V device has its own VCCIO pins and supports only one V_{CCIO}, either 1.2, 1.25, 1.35, 1.5, 1.8, or 3.0 V. The **LVDS** I/O standard is not supported when V_{CCIO} is 3.0 V. The LVDS input operations are supported when V_{CCIO} is 1.2, 1.25, 1.35, 1.5, 1.8, or 2.5 V. The **LVDS** output operations are only supported when V_{CCIO} is 2.5 V.

OCT Support and I/O Termination Schemes

Stratix V devices feature dynamic R_S and R_T OCT to provide I/O impedance matching and termination capabilities. OCT maintains signal quality, saves board space, and reduces external component costs.

Stratix V devices support the following OCT schemes:

- R_S with and without calibration
- R_T with calibration
- Dynamic R_S for single-ended I/O standards
- Dynamic R_T for single-ended I/O standards
- R_D for differential LVDS I/O standards

Stratix V devices support OCT in all I/O banks. You can use R_S and R_T OCT in the same I/O bank for different I/O standards if they use the same V_{CCIO} supply voltage. You can independently configure each I/O in an I/O bank to support R_S OCT, programmable current strength, or R_T OCT.

☞ You cannot configure both the R_S OCT and the programmable current strength for the same I/O buffer.

The Stratix V OCT calibration process uses the RZQ pin that is available in every calibration block in a given I/O bank for series- and parallel-calibrated termination. The RZQ pin shares the same V_{CCIO} supply with the I/O bank where it is located. It is a dual-purpose I/O pin and functions as a GPIO if you do not use the calibration circuit. When used for OCT calibration, the RZQ pin is connected to GND through an external 100- or 240-Ω reference resistor.

All I/O pins support calibrated R_S OCT, calibrated R_T OCT, and dynamic OCT for bidirectional pins. Dynamic R_T OCT is enabled for a bidirectional pin in receive mode and disabled in transmit mode.

The following connections are required to connect the RZQ pin through a reference resistor:

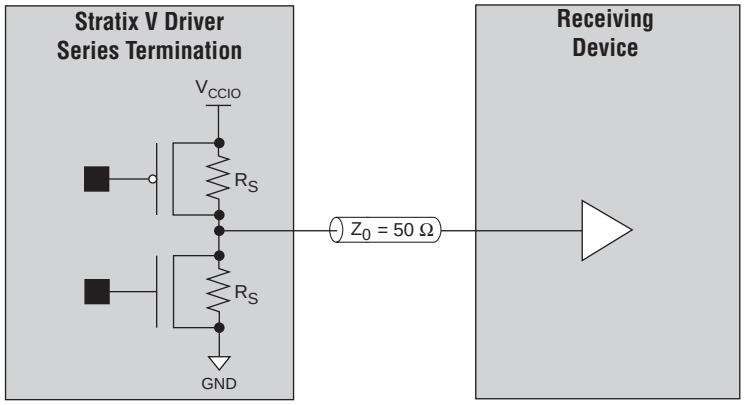
- RZQ pin is connected to GND through an external 240-Ω resistor for R_S OCT of 34, 40, 48, 60, and 80 Ω
- RZQ pin is connected to GND through an external 240-Ω resistor for R_T OCT of 20, 30, 40, 60, and 120 Ω
- RZQ pin is connected to GND through an external 100-Ω resistor for R_S OCT of 25 Ω and 50 Ω
- RZQ pin is connected to GND through an external 100-Ω resistor for R_T OCT of 50 Ω

R_S OCT Without Calibration

Stratix V devices support driver-impedance matching to provide the I/O driver with controlled output impedance that closely matches the impedance of the transmission line. As a result, you can significantly reduce reflections. Stratix V devices support R_S OCT for single-ended I/O standards (refer to Figure 5–3).

Figure 5–3 shows the R_S as the intrinsic impedance of the output transistors. Typical R_S values are 25 Ω and 50 Ω. When you select matching impedance, current strength is no longer selectable.

Figure 5–3. R_S OCT Without Calibration



To use OCT for the **SSTL Class I** I/O standard, you must select the 50-Ω R_S OCT setting, thus eliminating the external 25-Ω R_S (to match the 50-Ω transmission line). For the **SSTL Class II** I/O standard, you must select the 25-Ω R_S OCT setting (to match the 50-Ω transmission line and the near-end external 50-Ω pull-up to V_{TT}).

R_S OCT with Calibration

Stratix V devices support R_S OCT with calibration in all banks. The R_S OCT calibration circuit compares the total impedance of the I/O buffer to the external 240-Ω or 100-Ω reference resistor connected to the RZQ pin and dynamically enables or disables the transistors until they match.

The R_S shown in Figure 5–4 is the intrinsic impedance of the transistors. Calibration occurs at the end of device configuration. When the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers.

Figure 5–4. R_S OCT with Calibration

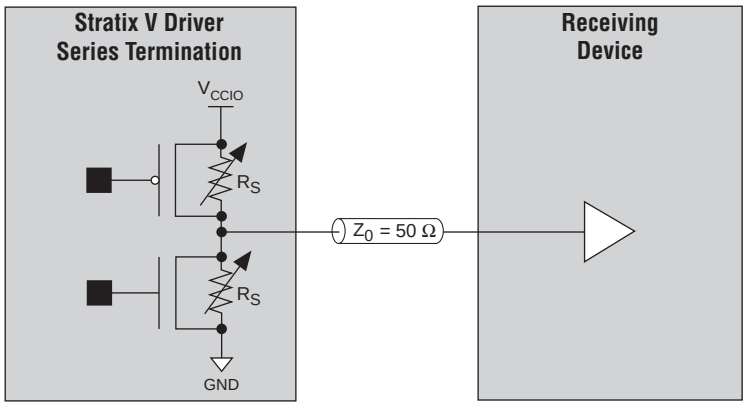


Table 5–8 lists the input and output termination for OCT with and without calibration on different I/O standards.

Table 5–8. Selectable I/O Standards for OCT With and Without Calibration (Part 1 of 2)

I/O standards	Output Termination		Input Termination	V_{REF} (V)	V_{CCIO} (V)
	Uncalibrated R_S OCT Setting, R_S (Ω)	Calibrated R_S OCT Setting, R_S (Ω) (1)	R_T OCT Setting, R_T (Ω)		
3.3V LVTTL/LVCMOS	25/50	25/50	—	—	3
2.5-V LVCMOS	25/50	25/50	—	—	2.5
1.8-V LVCMOS	25/50	25/50	—	—	1.8
1.5-V LVCMOS	25/50	25/50	—	—	1.5
1.2-V LVCMOS	25/50	25/50	—	—	1.2
SSTL-2 Class I	50	50	50	1.25	2.5
SSTL-2 Class II	25	25	50	1.25	2.5
SSTL-18 Class I	50	50	50	0.9	1.8
SSTL-18 Class II	25	25	50	0.9	1.8
SSTL15 Class I	50	50	50	0.75	1.5
SSTL15 Class II	25	25	50	0.75	1.5
HSTL 1.8 Class I	50	50	50	0.9	1.8
HSTL 1.8 Class II	25	25	50	0.9	1.8
HSTL 1.5 Class I	50	50	50	0.75	1.5
HSTL 1.5 Class II	25	25	50	0.75	1.5
HSTL 1.2 Class I	50	50	50	0.6	1.2
HSTL 1.2 Class II	25	25	50	0.6	1.2
SSTL15	25, 34, 40, 50	25, 34, 40, 50 (2)	20, 30, 40, 60, 120 (2)	0.75	1.5
SSTL135	34, 40	34, 40 (2)	20, 30, 40, 60, 120 (2)	0.675	1.35
SSTL125	34, 40	34, 40 (2)	20, 30, 40, 60, 120 (2)	0.625	1.25
SSTL12	40, 60, 240	40, 60, 240 (2)	60, 120 (2)	0.6	1.2
HSUL 1.2	34.3, 40, 48, 60, 80	34, 40, 48, 60, 80 (2)	(2)	0.6	1.2
Differential SSTL-2 Class I	50	50	50	1.25	2.5
Differential SSTL-2 Class II	25	25	50	1.25	2.5
Differential SSTL-18 Class I	50	50	50	0.9	1.8
Differential SSTL-18 Class II	25	25	50	0.9	1.8
Differential SSTL15 Class I	50	50	50	0.75	1.5
Differential SSTL15 Class II	25	25	50	0.75	1.5
Differential HSTL 1.8 Class I	50	50	50	0.9	1.8
Differential HSTL 1.8 Class II	25	25	50	0.9	1.8
Differential HSTL 1.5 Class I	50	50	50	0.75	1.5
Differential HSTL 1.5 Class II	25	25	50	0.75	1.5
Differential HSTL 1.2 Class I	50	50	50	0.6	1.2
Differential HSTL 1.2 Class II	25	25	50	0.6	1.2
Differential SSTL15	—	25, 34, 40, 50 (2)	20, 30, 40, 60, 120 (2)	—	1.5

Table 5–8. Selectable I/O Standards for OCT With and Without Calibration (Part 2 of 2)

I/O standards	Output Termination		Input Termination	V_{REF} (V)	V_{CCIO} (V)
	Uncalibrated R_S OCT Setting, R_S (Ω)	Calibrated R_S OCT Setting, R_S (Ω) (1)	R_T OCT Setting, R_T (Ω)		
Differential SSTL135	—	34, 40 (2)	20, 30, 40, 60, 120 (2)	—	1.35
Differential SSTL125	—	34, 40 (2)	20, 30, 40, 60, 120 (2)	—	1.25
Differential SSTL12	—	40, 60, 240 (2)	60, 120 (2)	—	1.2
Differential HSUL 1.2	—	34, 40, 48, 60, 80 (2)	(2)	—	1.2

Notes to Table 5–8:

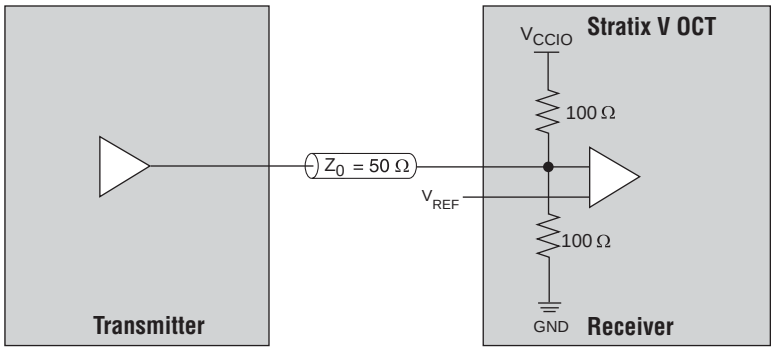
(1) The 25- and 50-Ω driver impedance are calibrated with the RZQ pin connected to a 100-Ω reference resistor to GND.

(2) The calibrated R_S and R_T OCT final values are pending silicon characterization.

R_T OCT with Calibration

Stratix V devices support R_T OCT with calibration in all banks. R_T OCT with calibration is only supported for input configuration of input and bidirectional pins. Output pin configurations do not support R_T OCT with calibration. Figure 5–5 shows R_T OCT with calibration. When you use R_T OCT, the V_{CCIO} of the bank must match the I/O standard of the pin where the R_T OCT is enabled.

Figure 5–5. R_T OCT with Calibration



The R_T OCT calibration circuit compares the total impedance of the I/O buffer to the external 100-Ω or 240-Ω resistors connected to the RZQ pin and dynamically enables or disables the transistors until they match. Calibration occurs at the end of device configuration. When the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers.

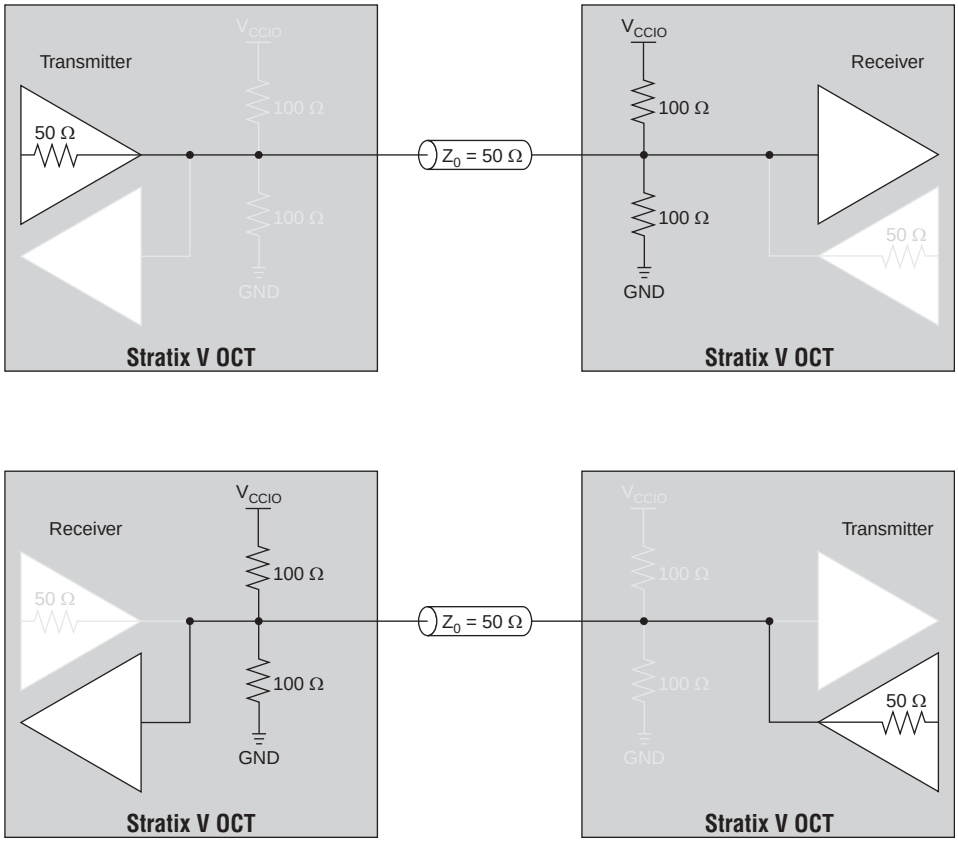
Dynamic OCT

Stratix V devices support dynamic R_S and R_T OCT for bidirectional I/Os in all I/O banks. Figure 5–6 shows the termination schemes supported in Stratix V devices. Dynamic R_T OCT is enabled only when the bidirectional I/O acts as a receiver and is disabled when it acts as a driver. Similarly, dynamic R_S OCT is enabled only when the bidirectional I/O acts as a driver and is disabled when it acts as a receiver. This feature is useful for terminating any high-performance bidirectional path because signal integrity is optimized depending on the direction of the data.

Altera recommends using the new I/O standards for the DDR3 memory interface with dynamic OCT schemes. These I/O standards save board space by reducing the number of external termination resistors used.

Using dynamic OCT also helps save power because device termination is internal instead of external. Termination only switches on during input operation, thus drawing less static power.

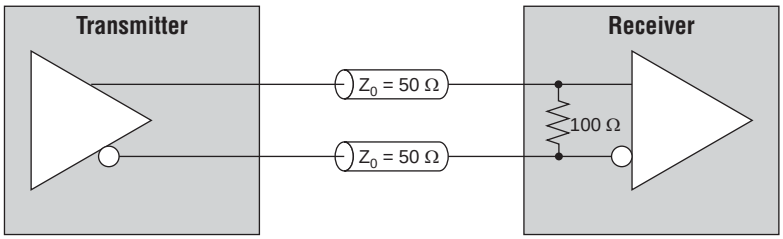
Figure 5–6. Dynamic R_T OCT in Stratix V Devices



LVDS Input R_D OCT

Stratix V devices support OCT for differential LVDS input buffers with a nominal resistance value of 100 Ω, as shown in Figure 5–7. R_D OCT is supported in all I/O banks. You can use R_D OCT when both the V_{CCIO} and V_{CCPD} is set to 2.5 V.

Figure 5–7. Differential Input OCT



Summary of OCT Assignments

Table 5–9 lists the OCT assignments for the Quartus II software.

Table 5–9. Summary of OCT Assignments in the Quartus II Software

Assignment Name	I/O Standard	Value (Ω)
Input Termination	**2.5-V SSTL Class I and II**	50
	1.8-V SSTL Class I and II	50
	1.5-V SSTL Class I and II	20, 25, 30, 40, 50, 60, 120
	1.35-V SSTL Class I	20, 30, 40, 60, 120
	1.25-V SSTL Class I	20, 30, 40, 60, 120
	1.2-V SSTL Class I	60, 120
	1.8-V HSTL Class I and II	50
	1.5-V HSTL Class I and II	50
	1.2-V HSTL Class I and II	50
	Input buffers for LVDS receivers *(1)*	Differential
Output Termination	**3.0-V LVTTL/CMOS**	25, 50
	2.5-V CMOS	25, 50
	1.8-V CMOS	25, 50
	1.5-V CMOS	25, 50
	1.2-V CMOS	25, 50
	2.5-V SSTL Class I	50
	2.5-V SSTL Class II	25
	1.8-V SSTL Class I	50
	1.8-V SSTL Class II	25
	1.5-V SSTL Class I	50
	1.5-V SSTL Class II	25
	1.5-V SSTL	34, 40
	1.35-V SSTL	34, 40
	1.25-V SSTL	34, 40
	1.2-V SSTL	40, 60, 240
	1.8-V HSTL Class I	50
	1.8-V HSTL Class II	25
	1.5-V HSTL Class I	50
	1.5-V HSTL Class II	25
	1.2-V HSTL Class I	50
	1.8-V HSTL Class II	25
	1.2-V HSUL	34.3, 40, 48, 60, 80

Note to Table 5–9:

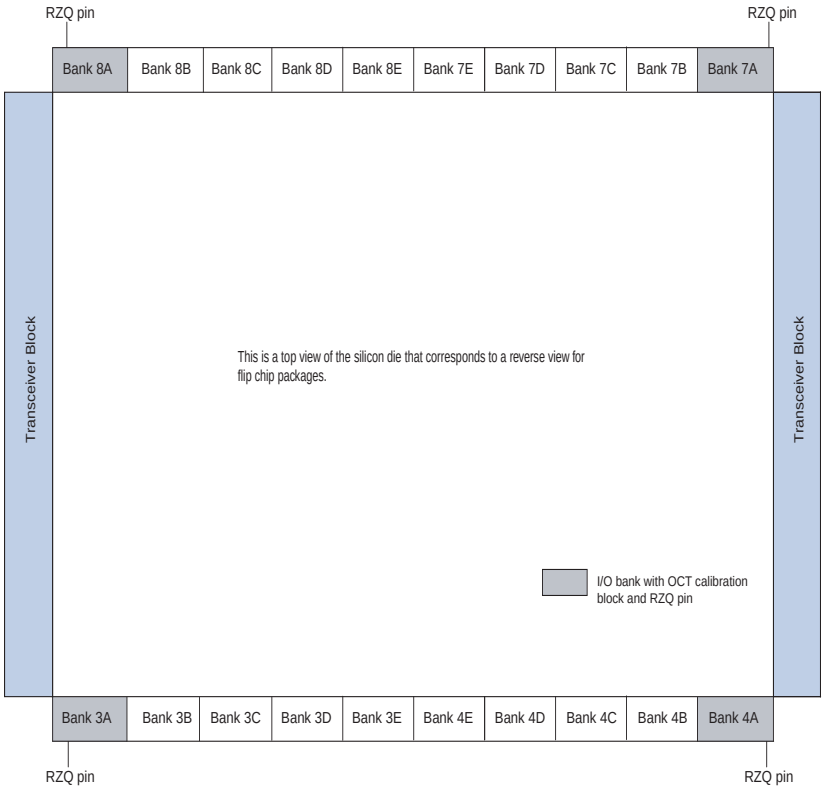
(1) You can enable differential OCT R_D in row I/O banks when both V_{CCIO} and V_{CCPD} are set to **2.5 V**.

OCT Calibration

Stratix V devices support calibrated R_S and calibrated R_T on all I/O pins with exceptions for dedicated configuration pins. You can calibrate using any of the available four to eight OCT calibration blocks, depending on the density of the device. Each calibration block contains one RZQ pin.

Figure 5–8 shows the location of I/O banks with OCT calibration blocks and RZQ pins.

Figure 5–8. OCT Calibration Block and RZQ Pin Location—Preliminary



Sharing an OCT Calibration Block on Multiple I/O Banks

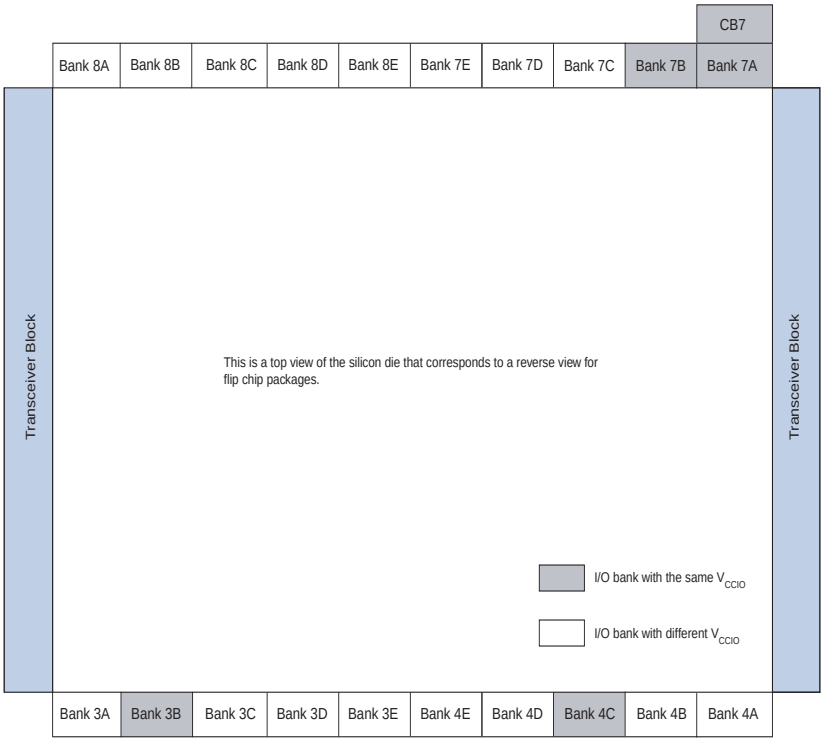
An OCT calibration block has the same V_{CCIO} as the I/O bank that contains the block. OCT calibration is supported on all I/O banks with different V_{CCIO} voltage standards, up to the number of available OCT calibration blocks. You can configure the I/O banks to receive calibration codes from any OCT calibration block with the same V_{CCIO}. All I/O banks with the same V_{CCIO} can share one OCT calibration block, even if that particular I/O bank has an OCT calibration block.

For example, Figure 5–9 shows a group of I/O banks that has the same V_{CCIO} voltage. If a group of I/O banks has the same V_{CCIO} voltage, you can use one OCT calibration block to calibrate the group of I/O banks placed around the periphery. Because 3B, 4C, 6C, and 7B have the same V_{CCIO} as bank 7A, you can calibrate all four I/O banks (3B, 4C, 6C, and 7B) with the OCT calibration block (CB7) located in bank 7A. You can enable this by serially shifting out the R_S OCT calibration codes from the OCT calibration block located in bank 7A to the I/O banks located around the periphery.

I/O banks that do not contain calibration blocks share calibration blocks with I/O banks that contains calibration blocks.

Figure 5–9 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. This figure does not show transceiver calibration blocks.

Figure 5–9. Example of Calibrating Multiple I/O Banks with One Shared OCT Calibration Block—Preliminary



OCT Calibration Block Modes of Operation

OCT calibration can occur in either power-up or user mode.

Power-Up Mode

In power-up mode, OCT calibration is automatically performed at power up. Calibration codes are shifted to selected I/O buffers before transitioning to user mode.

User Mode

In user mode, the OCTUSRCLK, ENAOCT, nCLRUSR, and ENASER signals are used to calibrate and serially transfer calibration codes from each OCT calibration block to any I/O.

Table 5–10 lists the user-controlled calibration block signal names and their descriptions.

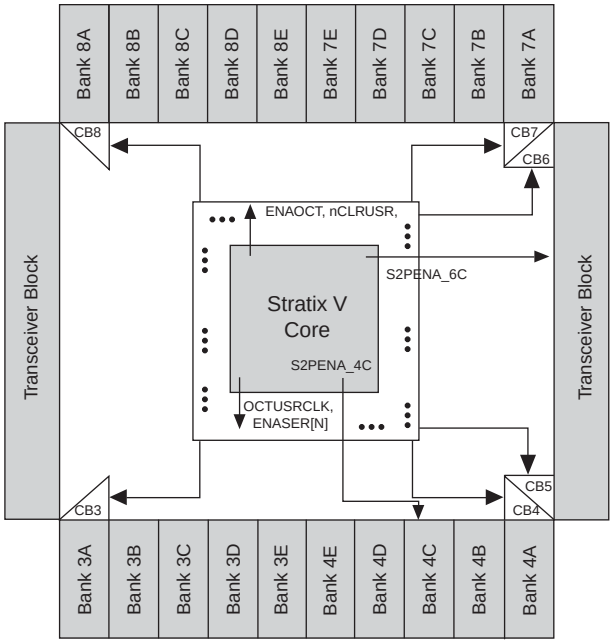
Table 5–10. OCT Calibration Block Ports for User Control

Signal Name	Description
OCTUSRCLK	Clock for OCT block.
ENAOCT	Enable OCT Calibration (Generated by user IP).
ENASER[7..0]	When ENOCT = 0, each signal enables the OCT serializer for the corresponding OCT calibration block. When ENAOCT = 1, each signal enables OCT calibration for the corresponding OCT calibration block.
S2PENA_<*bank#*>	Serial-to-parallel load enable per I/O bank.
nCLRUSR	Clear user.

Figure 5–10 shows the flow of the user signal. When ENAOCT is 1, all OCT calibration blocks are in calibration mode; when ENAOCT is 0, all OCT calibration blocks are in serial data transfer mode. The OCTUSRCLK clock frequency must be 20 MHz or less.

You must generate all user signals on the rising edge of the OCTUSRCLK signal.

Figure 5–10. Signals Used for User Mode Calibration



OCT Calibration

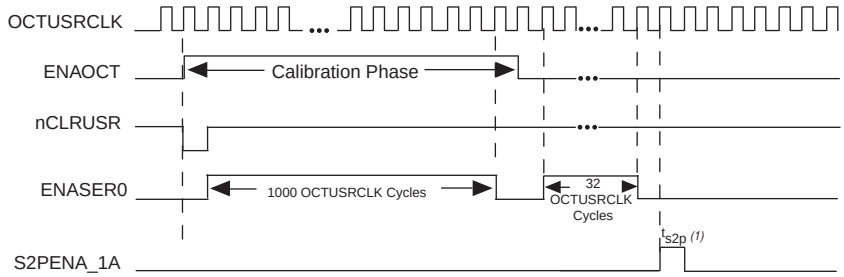
Figure 5–11 shows user mode signal-timing waveforms. To calibrate OCT block[N] (where N is a calibration block number), you must assert ENAOCT one cycle before asserting ENASER[N]. Also, nCLRUSR must be set to low for one OCTUSRCLK cycle before the ENASER[N] signal is asserted. Assert the ENASER[N] signals for 1000 OCTUSRCLK cycles to perform R_S OCT and R_T OCT calibration. You can deassert ENAOCT one clock cycle after the last ENASER is deasserted.

Serial Data Transfer

After you complete calibration, you must serially shift out the 32-bit OCT calibration codes (16-bit R_S OCT and 16-bit R_T OCT) from each OCT calibration block to the corresponding I/O buffers. Only one OCT calibration block can send out the codes at any time by asserting only one ENASER[N] signal at a time. After you deassert ENAOCT, wait at least one OCTUSRCLK cycle to enable any ENASER[N] signal to begin serial

transfer. To shift the 32-bit code from the OCT calibration block[N], you must assert ENASER[N] for exactly 32 OCTUSRCLK cycles. Between two consecutive asserted ENASER signals, there must be at least one OCTUSRCLK cycle gap (refer to Figure 5–11).

Figure 5–11. OCT User Mode Signal—Timing Waveform for One OCT Block



Note to Figure 5–11:

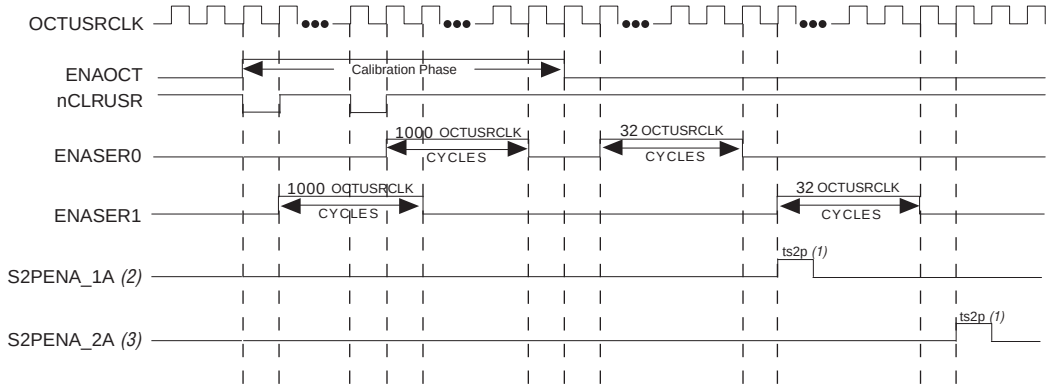
(1) $t_{s2p} \geq 25$ ns.

After calibrated codes are shifted in serially to each I/O bank, the calibrated codes must be converted from serial to parallel format before being used in the I/O buffers. Figure 5–11 shows the S2PENA signals that can be asserted at any time to update the calibration codes in each I/O bank. All I/O banks that received the codes from the same OCT calibration block can have S2PENA asserted at the same time, or at a different time, even while another OCT calibration block is calibrating and serially shifting codes. The S2PENA signal is asserted one OCTUSRCLK cycle after ENASER is deasserted for at least 25 ns. You cannot use I/Os for transmitting or receiving data when their S2PENA is asserted for parallel codes transfer.

Example of Using Multiple OCT Calibration Blocks

Figure 5–12 shows a signal timing waveform for two OCT calibration blocks doing R_S and R_T calibration. Calibration blocks can start calibrating at different times by asserting the ENASER signals at different times. ENAOCT must remain asserted while any calibration is ongoing. You must set nCLRUSR low for one OCTUSRCLK cycle before each ENASER[N] signal is asserted. In Figure 5–12, when you set nCLRUSR to 0 for the second time to initialize OCT calibration block 0, this does not affect OCT calibration block 1, whose calibration is already in progress.

Figure 5–12. OCT User-Mode Signal Timing Waveform for Two OCT Blocks



Notes to Figure 5–12:

(1) ts2p ≥ 25 ns

(2) S2PENA_1A is asserted in Bank 1A for calibration block 0.

(3) S2PENA_2A is asserted in Bank 2A for calibration block 1.

Termination Schemes for I/O Standards

The following sections describe the different termination schemes for the I/O standards used in Stratix V devices.

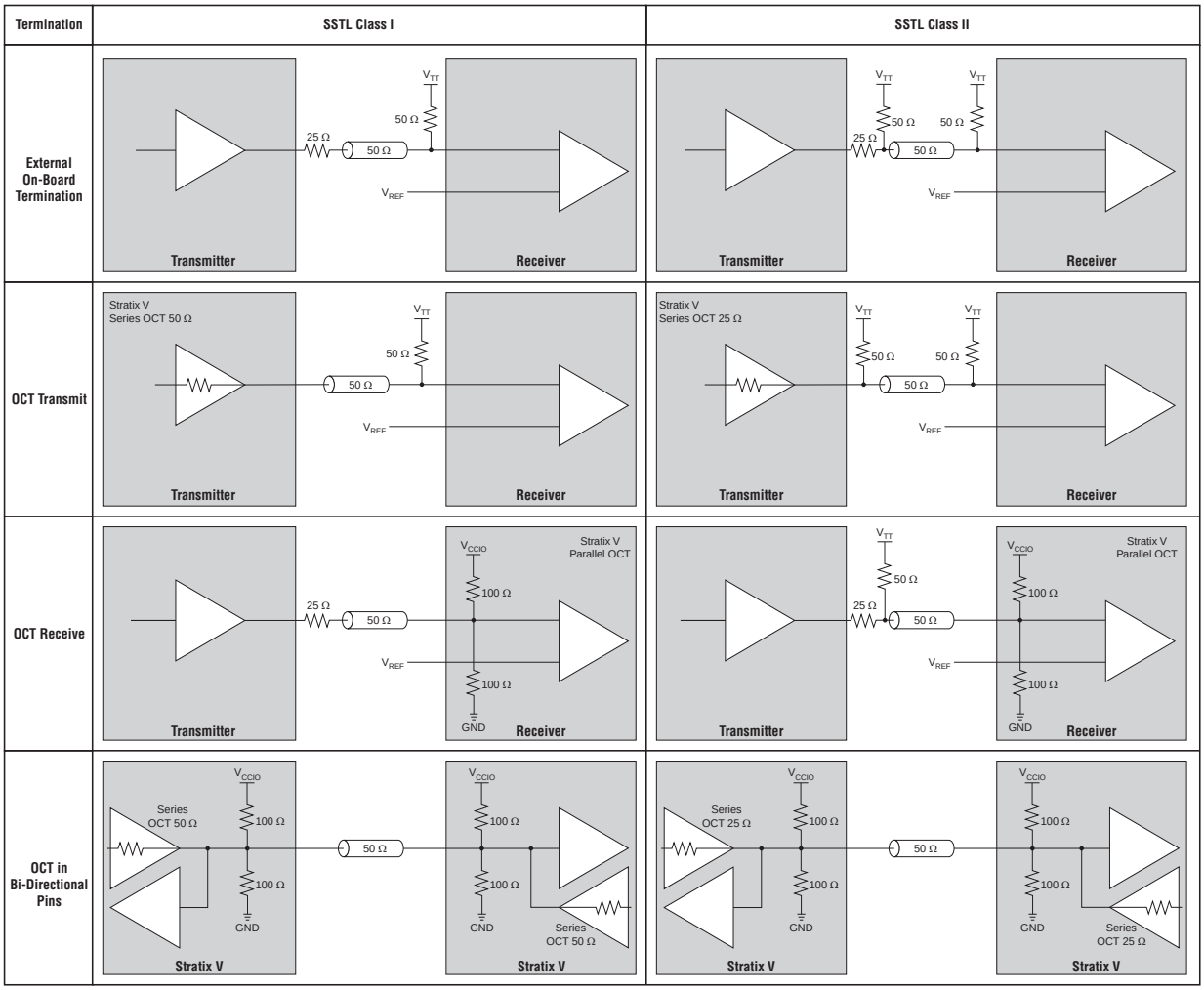
Single-Ended I/O Standards Termination

Voltage-referenced I/O standards require both an input reference voltage (V_{REF}) and a termination voltage (V_{TT}). The reference voltage of the receiving device tracks the termination voltage of the transmitting device.

The supported I/O standards such as **SSTL-15**, **SSTL-135**, **SSTL-125**, and **SSTL-12** typically do not require external board termination. Altera recommends using dynamic OCT with these I/O standards to save board space and cost by reducing the number of external termination resistors used.

Figure 5–13 shows the details of **SSTL** I/O termination on Stratix V devices.

Figure 5–13. SSTL I/O Standard Termination *(1)*

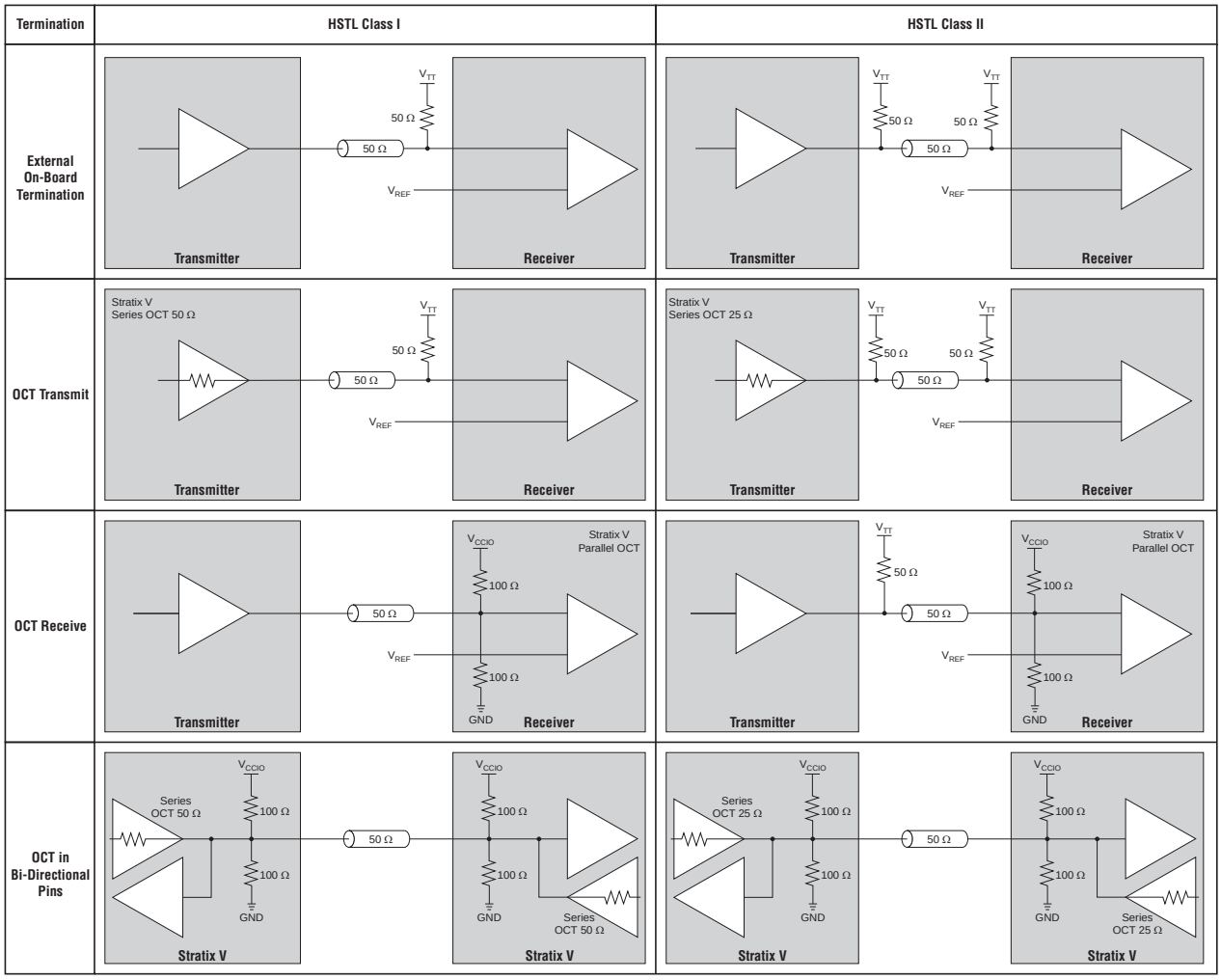


Note to Figure 5–13:

(1) This is not applicable for **SSTL-12**, **SSTL-15**, **SSTL-125,** and **SSTL-135** I/O standards.

Figure 5–14 shows the details of **HSTL** I/O termination on Stratix V devices.

Figure 5–14. HSTL I/O Standard Termination *(1)*



Note to Figure 5–14:

(1) This is not applicable for **HSUL-12** I/O standard.

You cannot use R_S and R_T OCT simultaneously. For more information, refer to "Dynamic OCT" on page 5–17.

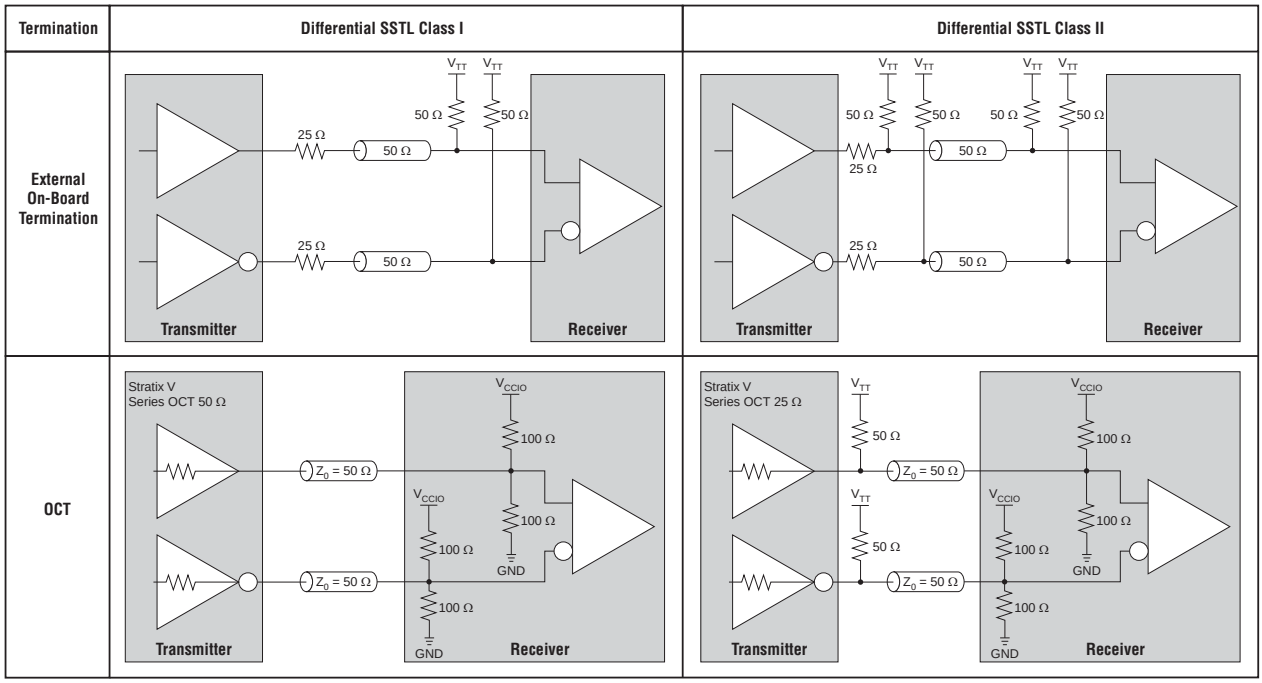
Differential I/O Standards Termination

Stratix V devices support **differential SSTL-18** and **SSTL-2**, **differential HSTL-18**, **HSTL-15**, **HSTL-12**, **LVDS**, **LVPECL**, **RSDS**, and **mini-LVDS** I/O standards. Figure 5–15 through Figure 5–21 show the details of various differential I/O terminations on these devices.

The supported I/O standards such as **differential SSTL-12**, d**ifferential SSTL-15**, **differential SSTL-125**, and **differential SSTL-135** typically do not require external board termination. Altera recommends using these I/O standards with dynamic OCT schemes to save board space and costs by reducing the number of external termination resistors used.

Differential HSTL, **SSTL**, and **HSUL** I/O standard outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted.

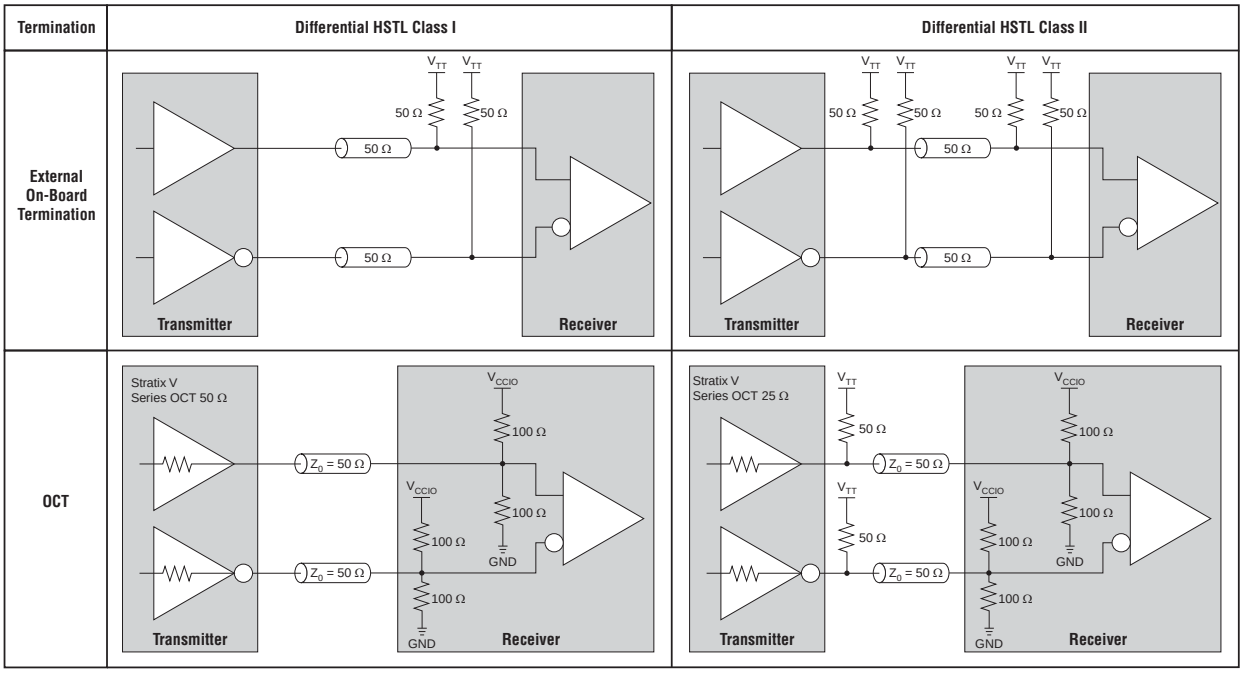
Figure 5–15. Differential SSTL I/O Standard Termination *(1)*



Note to Figure 5–15:

(1) This is not applicable for **differential SSTL-12**, **differential SSTL-15**, **differential SSTL-125**, **differential SSTL-135**, and **differential HSUL-12** I/O standards.

Figure 5–16. Differential HSTL I/O Standard Termination (1)



Note to Figure 5–16:

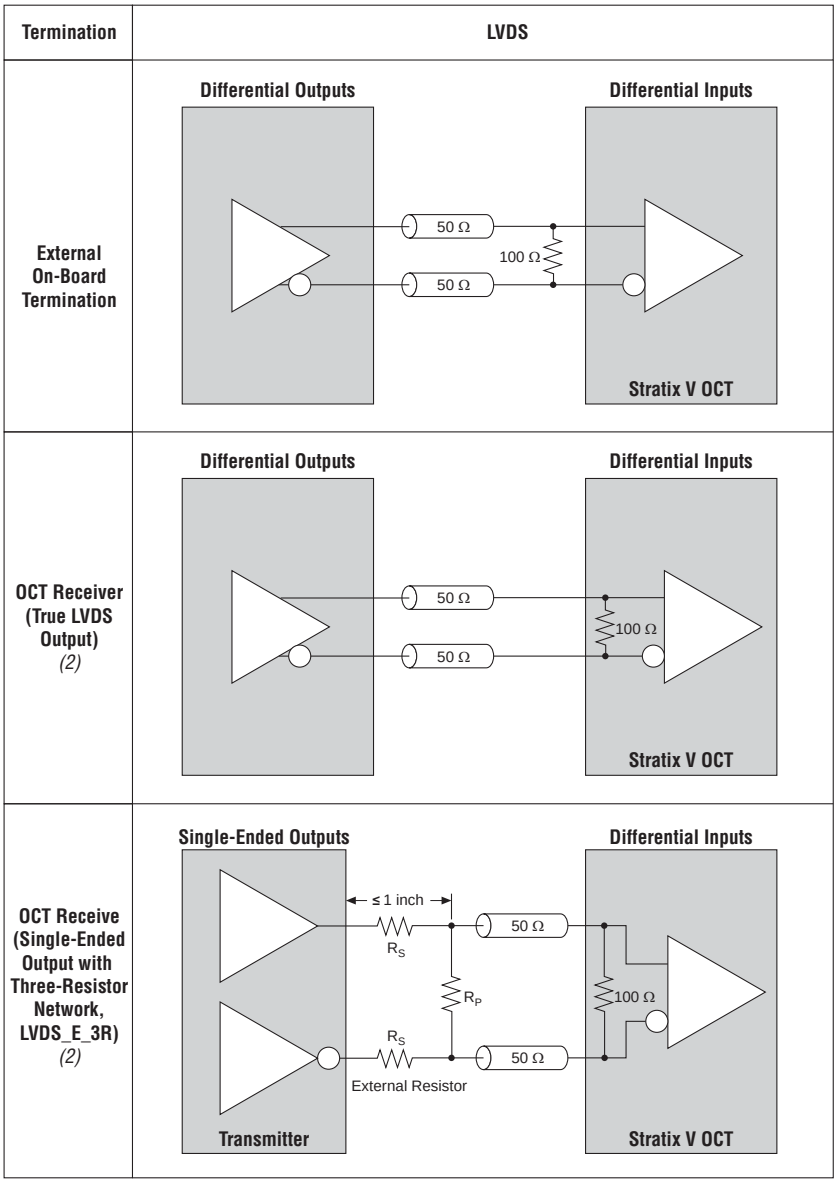
(1) This is not applicable for **differential HSUL-12** I/O standard.

LVDS

In Stratix V devices, the **LVDS** I/O standard requires a 2.5-V V_{CCIO} level. The **LVDS** input buffer requires 2.5-V V_{CCPD}. The **LVDS** receiver requires a 100-Ω termination resistor between the two signals at the input buffer. Stratix V devices provide an optional 100-Ω differential termination resistor in the device using R_D OCT when V_{CCIO} and V_{CCPD} are set to 2.5 V.

Figure 5–17 shows **LVDS** I/O standard termination. The on-chip differential resistor is available in all I/O banks.

Figure 5–17. LVDS I/O Standard Termination *(1)*



Notes to Figure 5–17:

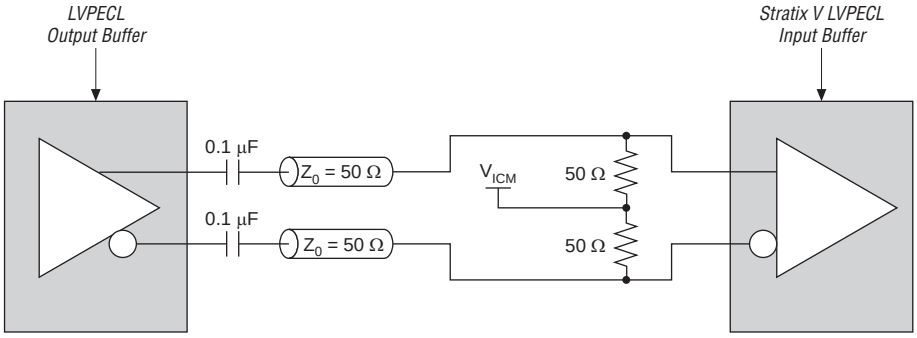
(1) For **LVDS** output with a three-resistor network, the R_S and R_P values are 120 Ω and 170 Ω, respectively. For **LVDS** output with one-resistor network, the R_P value is 120 Ω.

(2) All I/O banks support true LVDS output buffers and emulated **LVDS_E_3R** I/O standards. The emulated **LVDS** output buffers are configured with two single-ended output buffers and can be tri-stated.

Differential LVPECL

In Stratix V devices, the **LVPECL** I/O standard is supported on input clock pins on all the I/O banks. **LVPECL** output operation is not supported in Stratix V devices. **LVDS** input buffers are used to support **LVPECL** input operation. AC coupling is required when the **LVPECL** common-mode voltage of the output buffer is not matched to the **LVPECL** input common-mode voltage. Figure 5–18 shows the AC-coupled termination scheme. The 50-Ω resistors used at the receiver end are external to the device.

Figure 5–18. LVPECL AC-Coupled Termination *(1)*

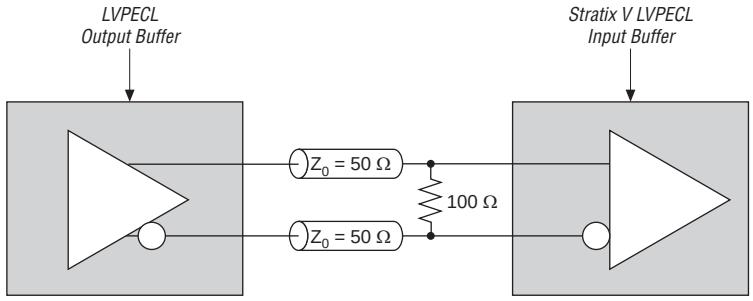


Note to Figure 5–18:

(1) The **LVPECL** AC/DC-coupled termination is applicable only when an Altera® FPGA transmitter is used.

DC-coupled **LVPECL** is supported if the **LVPECL** output common mode voltage is within the Stratix V **LVPECL** input buffer specification (refer to Figure 5–19).

Figure 5–19. LVPECL DC-Coupled Termination *(1)*



Note to Figure 5–19:

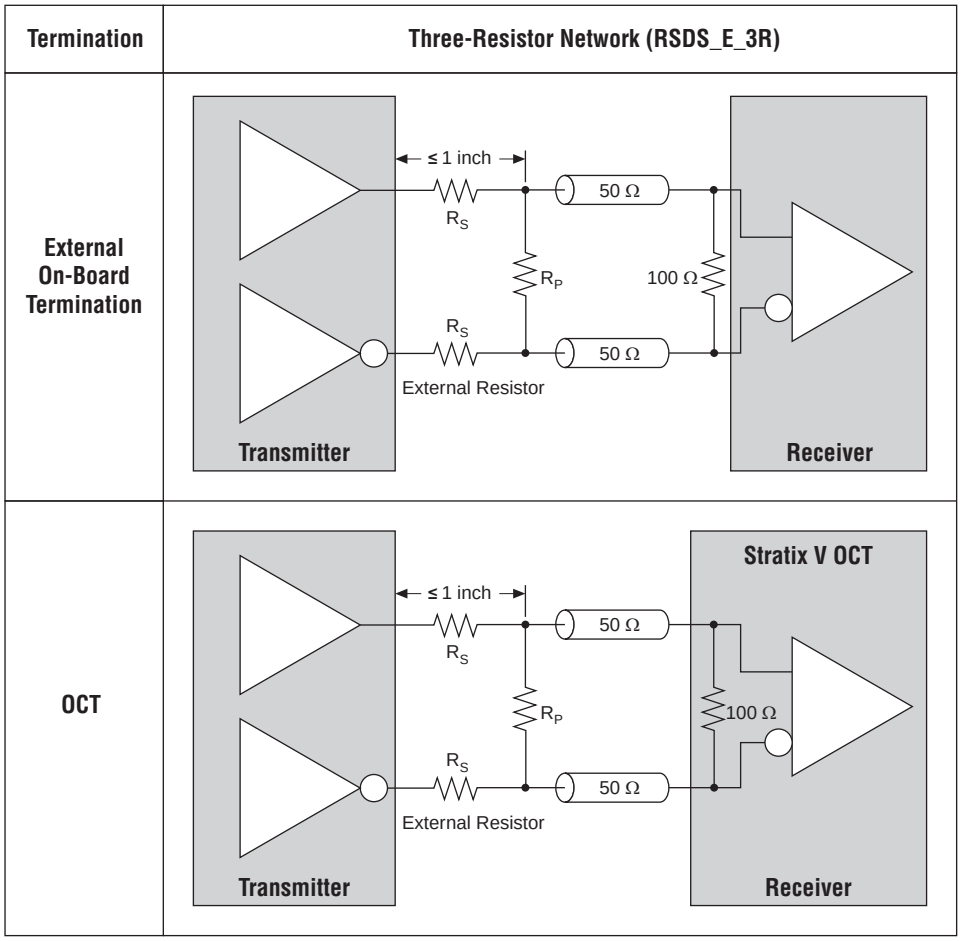
(1) The **LVPECL** AC/DC-coupled termination is applicable only when an Altera FPGA transmitter is used.

RSDS

Stratix V devices support the true **RSDS** output standard with data rates up to 360 Mbps using true **LVDS** output buffer types on all I/O banks. Emulated **RSDS** output buffers use two single-ended output buffers with external three-resistor networks and can be tri-stated. They are available in all I/O banks (refer to Figure 5–20).

Figure 5–20 shows the emulated **RSDS** I/O standard termination.

Figure 5–20. Emulated RSDS I/O Standard Termination *(1)*



Note to Figure 5–20:

(1) The R_S and R_P values are pending characterization.

A resistor network is required to attenuate the output-voltage swing to meet **RSDS** specifications. You can modify the three-resistor network values to reduce power or improve the noise margin. The resistor values chosen must satisfy Equation 5–1.

Equation 5–1.

$$\frac{R_s \times \frac{R_p}{2}}{R_s + \frac{R_p}{2}} = 50\Omega$$

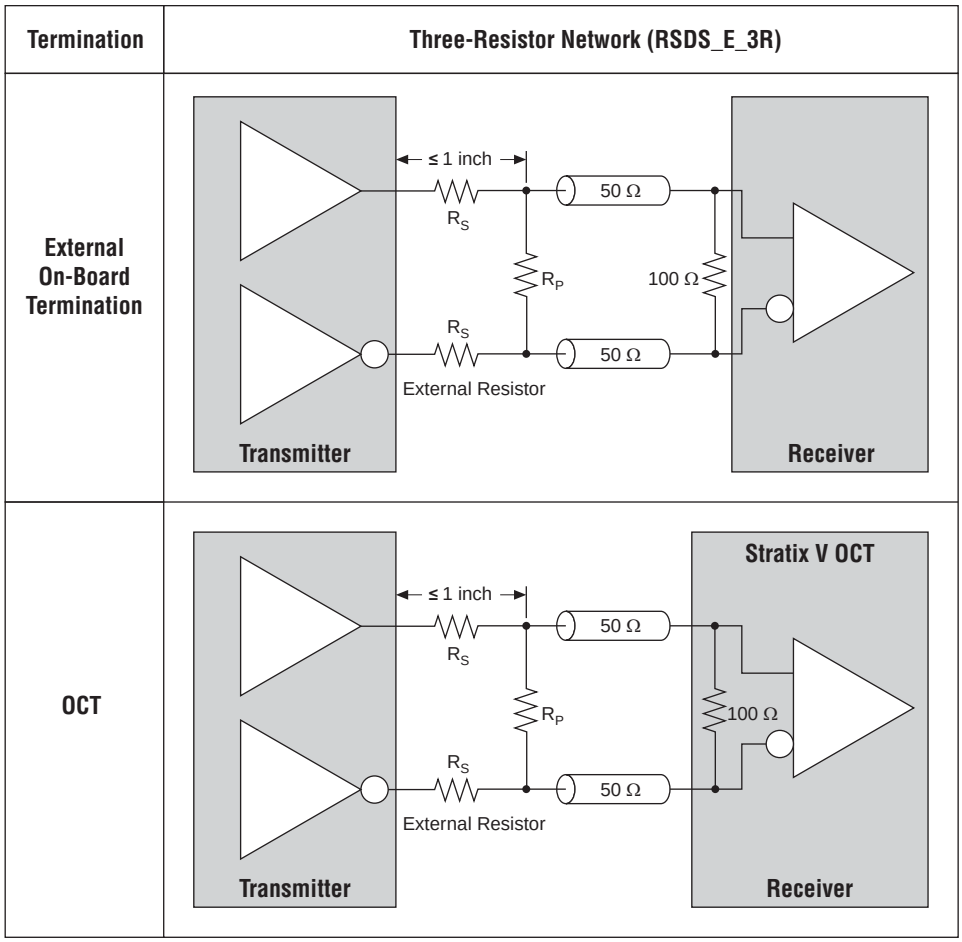
Altera recommends performing additional simulations using IBIS or SPICE models to validate that custom resistor values meet the **RSDS** I/O standard requirements.

For more information about the **RSDS** I/O standard, refer to the *RSDS Specification* available on the National Semiconductor website (www.national.com).

Mini-LVDS

Stratix V devices support the true **mini-LVDS** output standard with data rates up to 400 Mbps using **LVDS** output buffer types on all I/O banks. Emulated **mini-LVDS** output buffers use two single-ended output buffers with external three-resistor networks and can be tri-stated. They are available in all I/O banks (refer to Figure 5–21).

Figure 5–21. Emulated Mini-LVDS I/O Standard Termination *(1)*



Note to Figure 5–21:

(1) The R_S and R_P values are pending characterization.

A resistor network is required to attenuate the **LVDS** output voltage swing to meet the **mini-LVDS** specifications. You can modify the three-resistor network values to reduce power or improve noise margin. The resistor values chosen must satisfy Equation 5–1 on page 5–32.

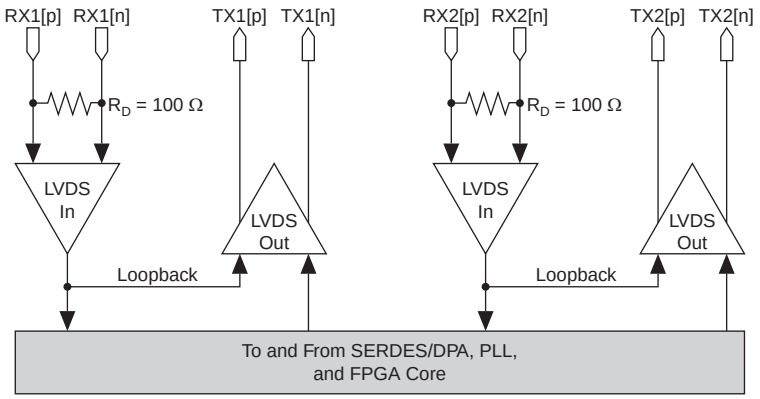
Altera recommends performing simulations using IBIS or SPICE models to validate that custom resistor R_S and R_P values meet the **mini-LVDS** requirements.

LVDS Direct Loopback Mode

LVDS direct loopback mode is available for true **LVDS** driver and receiver pairs in the I/O module that have the same pad group number. Figure 5–22 shows a datapath example of the **LVDS** direct loopback mode for I/O module 1 (RX1 and TX1) and I/O module 2 (RX2 and TX2). The data comes in from the receiver pin through the true **LVDS** input buffer and loops back to the true **LVDS** output buffer. You can use the **LVDS** direct loopback mode to verify the RX and TX buffers by checking the transmitted and received data.

☞ **LVDS** direct loopback mode is not supported for **LVDS** driver and receiver pairs from different I/O modules (for example, between RX1 and TX2).

Figure 5–22. LVDS Direct Loopback Path



To use the loopback mode, you must reset and recompile your existing design. The Quartus II software generates an error if the specified I/O standard and pin direction do not meet the requirements for direct loopback mode.

For an **LVDS** output pair already in use, you can apply the **LVDS** direct loopback mode to override the connection from the device core with the signal from the true differential input buffer in the same I/O module. Use this mode to observe the true **LVDS** input signal of a completed design.

Design Considerations

Although Stratix V devices feature various I/O capabilities for high-performance and high-speed system designs, there are several other design considerations that require your attention to ensure the success of your designs.

I/O Bank Restrictions

Each I/O bank can simultaneously support multiple I/O standards. The following sections provide guidelines for mixing non-voltage-referenced and voltage-referenced I/O standards in Stratix V devices.

Non-Voltage-Referenced Standards

Each I/O bank of a Stratix V device has its own VCCIO pins and supports only one V_{CCIO}, either 1.2, 1.25, 1.35, 1.5, 1.8, 2.5, or 3.0 V. An I/O bank can simultaneously support any number of input signals with different I/O standard assignments if it meets the V_{CCIO} and V_{CCPD} requirement (refer to Table 5–2 on page 5–3).

For output signals, a single I/O bank supports non-voltage-referenced output signals that are driving at the same voltage as V_{CCIO}. Because an I/O bank can only have one V_{CCIO} value, it can only drive out that one value for non-voltage-referenced signals. For example, an I/O bank with a 2.5-V V_{CCIO} setting can support 2.5-V standard inputs and outputs as well as **3.0-V LVCMOS** inputs (but not **3.0-V LVCMOS** output or bidirectional pins).

Voltage-Referenced Standards

To accommodate voltage-referenced I/O standards, each Stratix V device's I/O bank supports multiple VREF pins feeding a common V_{REF} bus. The number of available VREF pins increases as device density increases. If these pins are not used as VREF pins, they cannot be used as GPIO pins and must be tied to V_{CCIO} or GND. Each bank can only have a single V_{CCIO} voltage level and a single V_{REF} voltage level at a given time.

An I/O bank featuring single-ended or differential standards can support voltage-referenced standards if all voltage-referenced standards use the same V_{REF} setting.

For performance reasons, voltage-referenced input standards use their own V_{CCPD} level as the power source. This feature allows you to place voltage-referenced input signals in an I/O bank with a V_{CCIO} of 2.5 V or below. For example, you can place **HSTL-15** input pins in an I/O bank with 2.5-V V_{CCIO}. However, the voltage-referenced input with R_T OCT enabled requires the V_{CCIO} of the I/O bank to match the voltage of the input standard. R_T OCT cannot be supported for the **HSTL-15** I/O standard when V_{CCIO} is 2.5 V.

Voltage-referenced bidirectional and output signals must be the same as the I/O bank's V_{CCIO} voltage. For example, you can only place **SSTL-2** output pins in an I/O bank with a 2.5-V V_{CCIO}.

Mixing Voltage-Referenced and Non-Voltage-Referenced Standards

An I/O bank can support both voltage-referenced and non-voltage-referenced pins by applying each of the rule sets individually. For example, an I/O bank can support **SSTL-18** inputs and 1.8-V inputs and outputs with a 1.8-V V_{CCIO} and a 0.9-V V_{REF}. Similarly, an I/O bank can support 1.5-V standards, 1.8-V inputs (but not outputs), and **HSTL** and **HSTL-15** I/O standards with a 1.5-V V_{CCIO} and 0.75-V V_{REF}.

V_{CCPD} Restriction

One VCCPD pin is shared in the particular group of I/O banks. For example, the I/O banks with the same number 7A, 7B, 7C, and 7D form a group that share the same VCCPD pin. This is true for all I/O banks except for I/O banks 3A, 3B, 3C, 3D, and 3E. Banks 3A and 3B form a group with one VCCPD pin while banks 3C, 3D, and 3E form a different group with its own VCCPD pin.

Not all Stratix V device packages have bank 3E.

If one I/O bank is using 3.0-V V_{CCPD}, other I/O banks in the same group must also use 3.0-V V_{CCPD}.

If you are using an output or bidirectional pin with the **3.3 V-LVTTL/LVCMOS** I/O standard, you must adhere to this restriction manually with location assignments.

LVDS Channels

To drive the LVDS channels, you must use the PLLs in integer PLL mode.

For more information about the LVDS channels, refer to the *High-Speed Differential I/O Interfaces and DPA in Stratix V Devices* chapter.

Document Revision History

Table 5–11 lists the revision history for this chapter.

Table 5–11. Document Revision History

Date	Version	Changes
June 2012	1.5	■ Added "Summary of OCT Assignments" and "LVDS Channels" sections. ■ Updated Table 5–2, Table 5–3, Table 5–4, Table 5–5, and Table 5–8. ■ Updated "Pull-Up Resistor", "Differential Output Voltage", and "Programmable IOE Delay" sections.
November 2011	1.4	■ Updated Figure 5–2. ■ Updated Table 5–3, Table 5–4, and Table 5–5.
May 2011	1.3	■ Chapter moved to volume 2 for the 11.0 release. ■ Added Table 5–4, Table 5–5, Table 5–6, Table 5–7, and Table 5–8. ■ Updated "Single-Ended I/O Standards Termination", "Differential I/O Standards Termination", and "V_{CCPD} Restriction" sections. ■ Updated Table 5–3 and Table 5–11. ■ Updated Figure 5–1, Figure 5–8, Figure 5–9, Figure 5–10, Figure 5–17, Figure 5–20, and Figure 5–21. ■ Minor text edits.
January 2011	1.2	Updated Table 5–2.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

6. High-Speed Differential I/O Interfaces and DPA in Stratix V Devices

SV51007-1.4

This chapter describes the significant advantages of the high-speed differential I/O interfaces and the dynamic phase aligner (DPA) over single-ended I/Os and their contribution to the overall system bandwidth achievable with Stratix® V FPGAs.

The following sections describe the Stratix V high-speed differential I/O interfaces and DPA in detail:

- "Locations of the I/O Banks" on page 6–3
- "LVDS Channels" on page 6–3
- "LVDS SERDES" on page 6–6
- "Differential Transmitter" on page 6–7
- "Differential Receiver" on page 6–12
- "LVDS Interface with the Use External PLL Option Enabled" on page 6–23
- "PLLs and Stratix V Clocking" on page 6–27
- "Source-Synchronous Timing Budget" on page 6–28
- "Differential Pin Placement Guidelines" on page 6–35

Overview

All Stratix V devices have built-in serializer/deserializer (SERDES) circuitry that supports high-speed LVDS interfaces at data rates of up to 1.434 Gbps. SERDES circuitry is configurable to support source-synchronous communication protocols such as RapidIO™, XSBI, serial peripheral interface (SPI), and asynchronous protocols such as SGMII and Gigabit Ethernet (GbE).

The Stratix V device family has the following dedicated circuitry for high-speed differential I/O support:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer
- Data realignment
- DPA
- Synchronizer (FIFO buffer)
- Phase-locked loops (PLLs)

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

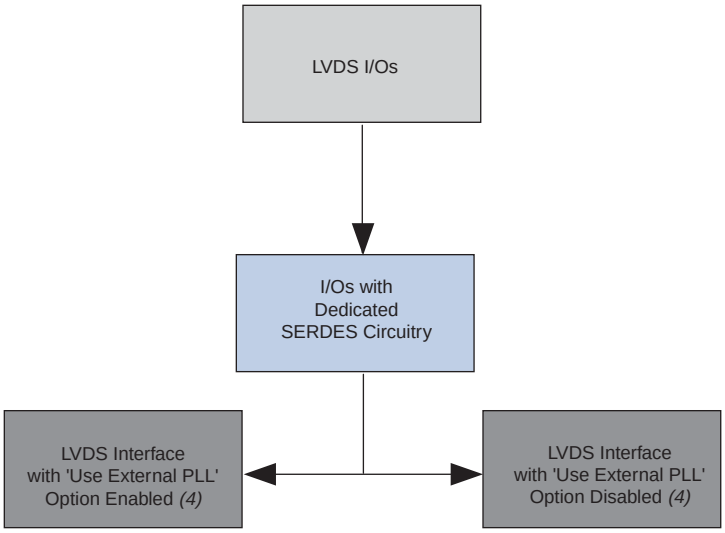
ISO 9001:2008 Registered

For high-speed differential interfaces, the Stratix V device family supports the following differential I/O standards:

- **LVDS**
- **Mini-LVDS**
- Reduced swing differential signaling (**RSDS**)

Stratix V devices support ubiquitous I/Os. Row and column I/Os support the same features. Figure 6–1 shows I/O bank support for the Stratix V device family.

Figure 6–1. I/O Bank Support in the Stratix V Device Family *(1), (2), (3)*


LVDS I/Os
I/Os with Dedicated SERDES Circuitry
LVDS Interface with 'Use External PLL' Option Enabled (4)
LVDS Interface with 'Use External PLL' Option Disabled (4)

Notes to Figure 6–1:

(1) All I/O banks support **LVDS**, **RSDS**, and **mini-LVDS** I/O standards using true LVDS output buffers without resistor networks.

(2) All I/O banks support **RSDS** and **mini-LVDS** I/O standards using two single-ended output buffers with a three-resistor (RSDS_E_3R and mini-LVDS_E_3R) network.

(3) 100-Ω differential input termination (R_D OCT) is supported in all I/O banks.

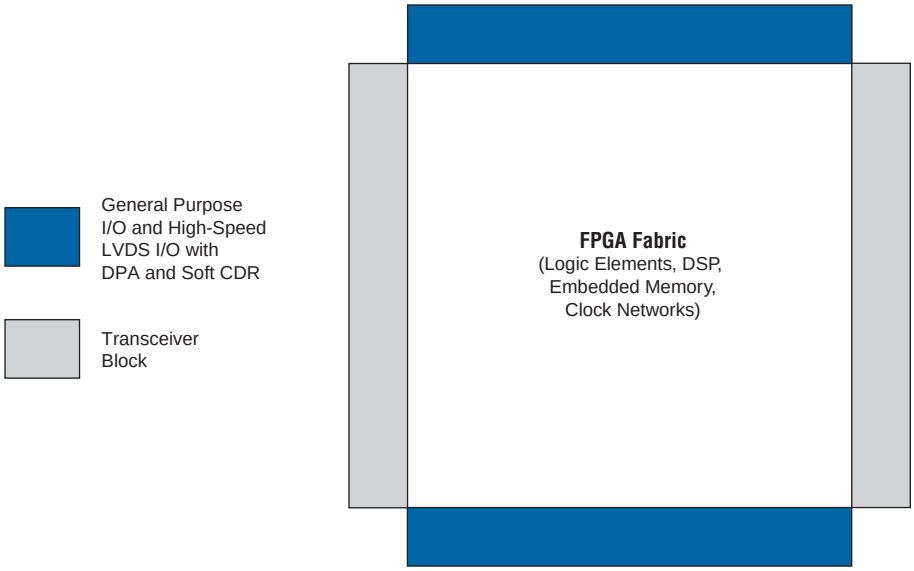
(4) The ALTLVDS **Use External PLL** option will only be available in future Quartus II software releases.

The ALTLVDS transmitter and receiver requires various clock and load enable signals from a fractional PLL. The Quartus® II software configures the PLL settings automatically. It is also responsible for generating the various clock and load enable signals based on the input reference clock and selected data rate.

Locations of the I/O Banks

Stratix V devices contain up to 26 I/O banks. The dedicated circuitry that supports high-speed differential I/Os is located in the top and bottom banks. Figure 6–2 shows the high-level SERDES/DPA location in the Stratix V devices.

Figure 6–2. High-Speed Differential I/Os with DPA Locations for Stratix V Devices


General Purpose I/O and High-Speed LVDS I/O with DPA and Soft CDR
Transceiver Block
FPGA Fabric
(Logic Elements, DSP, Embedded Memory, Clock Networks)

LVDS Channels

The Stratix V device family supports LVDS on all I/O banks. Both row and column I/Os support true LVDS input buffers with R_D OCT and true LVDS output buffers. Alternatively, you can configure the LVDS pins as emulated LVDS output buffers that use two single-ended output buffers with an external resistor network to support **LVDS**, **mini-LVDS**, and **RSDS** standards. Stratix V devices offer single-ended I/O reference clock support for the **LVDS**.

Stratix V devices support dedicated SERDES and DPA circuitry. For the supported I/O banks, refer to Figure 6–2.

☞ Emulated differential output buffers support tri-state capability.

Table 6–1 through Table 6–4 list the maximum number of LVDS I/Os supported in Stratix V devices. You can design the LVDS I/Os as true LVDS buffers or emulated LVDS buffers, as long as the combination of the two do not exceed the maximum count.

Table 6–1. LVDS Channels Supported in 5SGXA3, 5SGXA4, 5SGXA5, 5SGXA7, and 5SGXA9 Devices (1), (2)—Preliminary

Device	5SGXA3				5SGXA4				5SGXA5				5SGXA7				5SGXA9			
I/O Bank Side	Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom	
Clock Region (3)	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
True LVDS Input Buffers, RX	42	42	45	42	45	42	45	42	54	51	54	51	54	51	54	51	54	51	54	51
True LVDS Output Buffers, TX	45	42	45	42	45	42	45	42	54	51	54	51	54	51	54	51	54	51	54	51
Emulated LVDS Output buffers, eTX	90	84	90	84	90	84	90	84	108	102	108	102	108	102	108	102	108	102	108	102

Notes to Table 6–1:

(1) The LVDS channel count does not include dedicated clock input pins.

(2) The number of LVDS receiver (RX) and transmitter (TX) channels are preliminary.

(3) For more information about the clock region, refer to the "PLLs and Stratix V Clocking" on page 6–27.

Table 6–2. LVDS Channels Supported in 5SGXAB, 5SGXB5, 5SGXB6, 5SGXB9, and 5SGXBB Devices (1), (2)—Preliminary

Device	5SGXAB				5SGXB5				5SGXB6				5SGXB9				5SGXBB			
I/O Bank Side	Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom	
Clock Region (3)	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
True LVDS Input Buffers, RX	54	51	54	51	42	39	42	39	42	39	42	39	54	51	54	51	54	51	54	51
True LVDS Output Buffers, TX	54	51	54	51	42	39	42	39	42	39	42	39	54	51	54	51	54	51	54	51
Emulated LVDS Output buffers, eTX	108	102	108	102	84	78	84	78	84	78	84	78	108	102	108	102	108	102	108	102

Notes to Table 6–2:

(1) The LVDS channel count does not include dedicated clock input pins.

(2) The number of LVDS receiver (RX) and transmitter (TX) channels are preliminary.

(3) For more information about the clock region, refer to the "PLLs and Stratix V Clocking" on page 6–27.

Table 6–3. LVDS Channels Supported in Stratix V GS Devices (1), (2)—Preliminary

Device	5SGSD3				5SGSD4				5SGSD5				5SGSD6						5SGSD8					
I/O Bank Side	Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom		Right		Top		Bottom		Right	
Clock Region (3)	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Top	Bottom	Left	Right	Left	Right	Top	Bottom
True LVDS Input Buffers, RX	42	39	42	39	45	42	45	42	45	42	45	42	44	45	38	45	25	26	44	45	38	45	25	26
True LVDS Output Buffers, TX	42	39	42	39	45	42	45	42	45	42	45	42	50	47	44	47	31	32	50	47	44	47	31	32
Emulated LVDS Output buffers, eTX	84	78	84	78	90	84	90	84	90	84	90	84	94	92	82	92	56	58	94	92	82	92	56	58

Notes to Table 6–3:

(1) The LVDS channel count does not include dedicated clock input pins.
(2) The number of LVDS receiver (RX) and transmitter (TX) channels are preliminary.
(3) For more information about the clock region, refer to the "PLLs and Stratix V Clocking" on page 6–27.

Table 6–4. LVDS Channels Supported in Stratix V GT and E Devices (1), (2)—Preliminary

Device	5SGTC5				5SGTC7				5SEE9				5SEEB			
I/O Bank Side	Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom	
Clock Region (3)	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
True LVDS Input Buffers, RX	54	51	54	51	54	51	54	51	54	51	54	51	54	51	54	51
True LVDS Output Buffers, TX	54	51	54	51	54	51	54	51	54	51	54	51	54	51	54	51
Emulated LVDS Output buffers, eTX	108	102	108	102	108	102	108	102	108	102	108	102	108	102	108	102

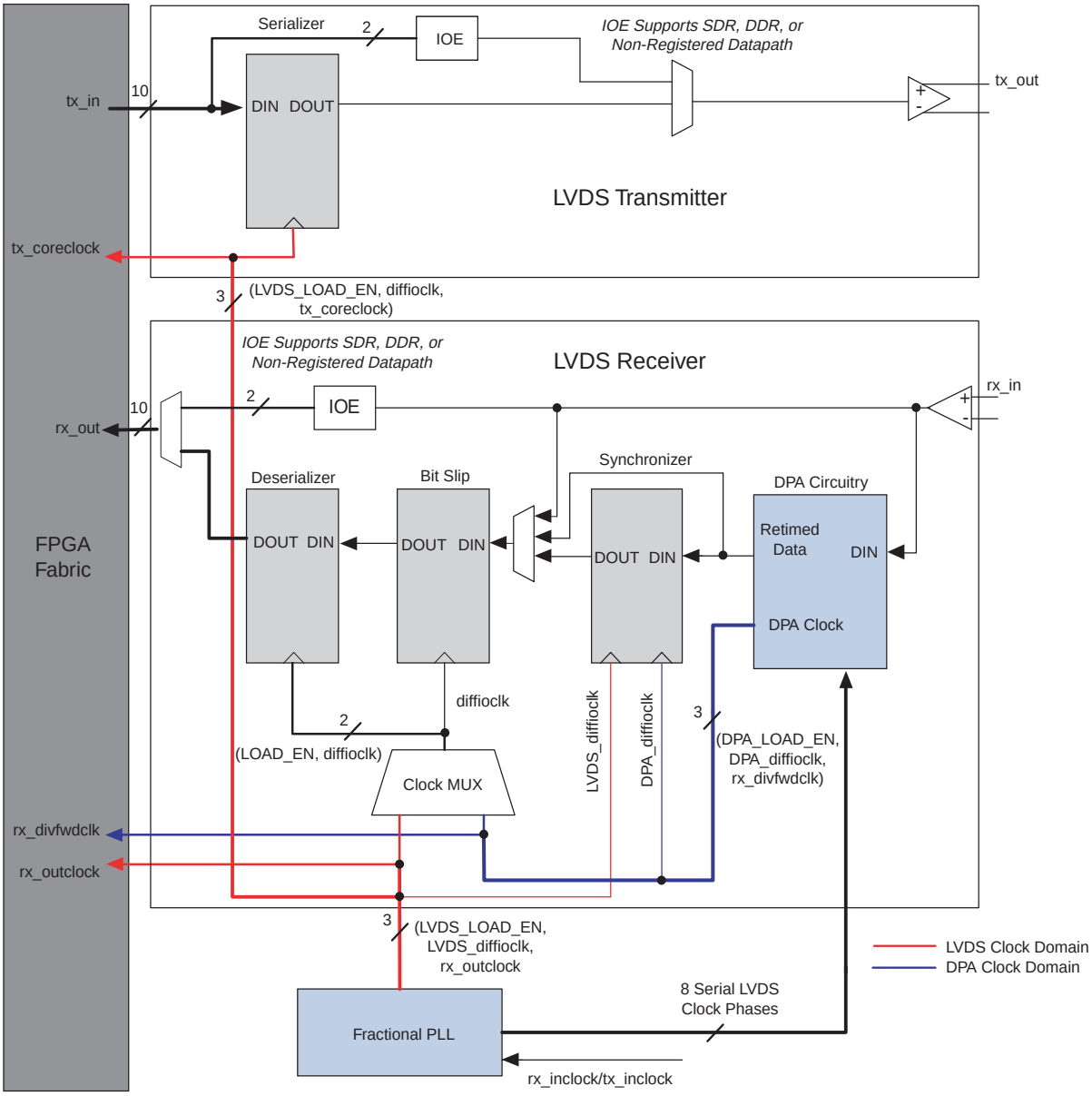
Notes to Table 6–4:

(1) The LVDS channel count does not include dedicated clock input pins.
(2) The number of LVDS receiver (RX) and transmitter (TX) channels are preliminary.
(3) For more information about the clock region, refer to the "PLLs and Stratix V Clocking" on page 6–27.

LVDS SERDES

Figure 6–3 shows a transmitter and receiver block diagram for the LVDS SERDES circuitry. This diagram shows the interface signals of the transmitter and receiver datapath. For more information, refer to "Differential Transmitter" on page 6–7 and "Differential Receiver" on page 6–12.

Figure 6–3. LVDS SERDES *(1), (2), (3)*



Notes to Figure 6–3:

(1) This diagram shows a shared PLL between the transmitter and receiver. If the transmitter and receiver are not sharing the same PLL, two fractional PLLs are required.

(2) In single data rate (SDR) and double data rate (DDR) modes, the data width is 1 and 2 bits, respectively.

(3) The `tx_in` and `rx_out` ports have a maximum data width of 10 bits.

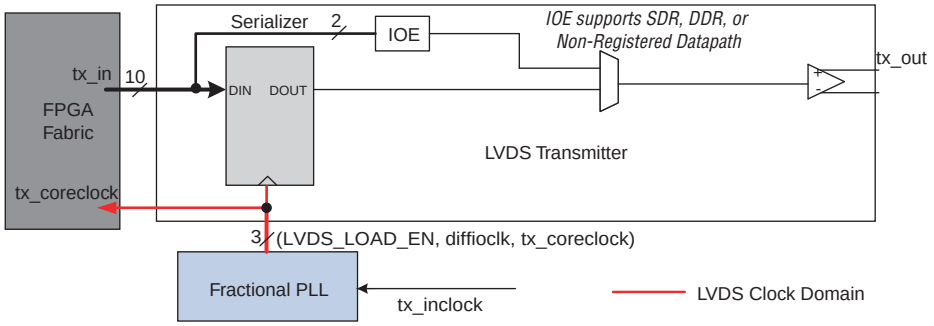
For more information about the LVDS transmitter and receiver port list and settings using ALTLVDS, refer to the *ALTLVDS Megafunction User Guide*.

Differential Transmitter

The Stratix V transmitter has dedicated circuitry to provide support for LVDS signaling. The dedicated circuitry consists of a differential buffer, a serializer, and fractional PLLs that can be shared between the transmitter and receiver. The differential buffer can drive out LVDS, mini-LVDS, and RSDS signaling levels. The serializer takes up to 10 bits wide parallel data from the FPGA fabric, clocks it into the load registers, and serializes it using shift registers clocked by the fractional PLL before sending the data to the differential buffer. The MSB of the parallel data is transmitted first.

The load enable (`LVDS_LOAD_EN`) signal and the `diffioclk` signal (the clock running at serial data rate) generated from the fractional PLL clocks the load and shift registers. You can statically set the serialization factor to x3, x4, x5, x6, x7, x8, x9, or x10 using the Quartus II software. The load enable signal is derived from the serialization factor setting. Figure 6–4 shows a block diagram of the Stratix V transmitter.

Figure 6–4. Stratix V Transmitter *(1), (2)*



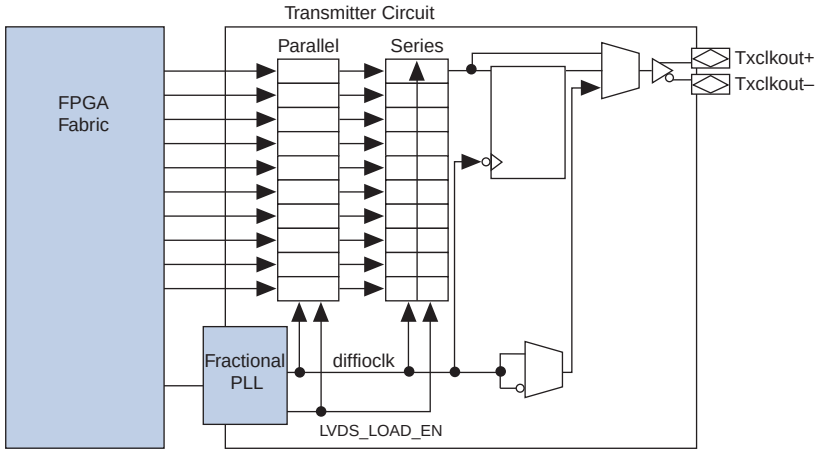
Notes to Figure 6–4:

(1) In SDR and DDR modes, the data width is 1 and 2 bits, respectively.
(2) The `tx_in` port has a maximum data width of 10 bits.

You can configure any Stratix V transmitter data channel to generate a source-synchronous transmitter clock output. This flexibility allows the placement of the output clock near the data outputs to simplify board layout and reduce clock-to-data skew. Different applications often require specific clock-to-data alignments or specific data-rate-to-clock-rate factors. The transmitter can output a clock signal at the same rate as the data with a maximum frequency of 717 MHz. The output clock can also be divided by a factor of 1, 2, 4, 6, 8, or 10, depending on the serialization factor. You can set the phase of the clock in relation to the data at 0° or 180° (edge or center aligned). The fractional PLLs provide additional support for other phase shifts in 45° increments. These settings are made statically in the

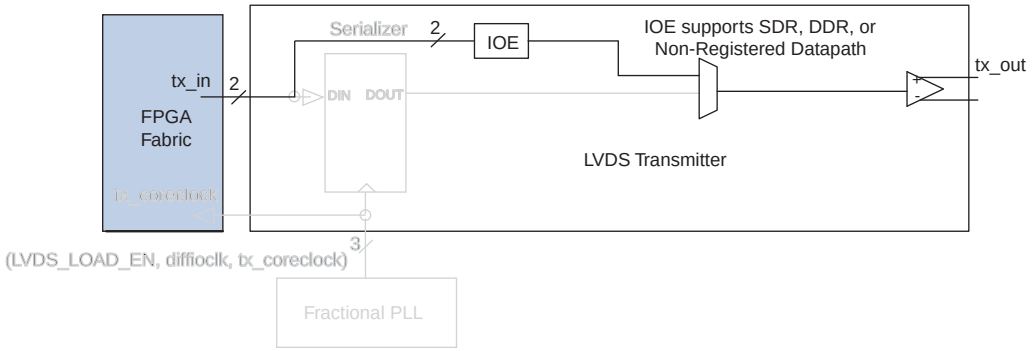
Quartus II MegaWizard™ Plug-In Manager. Figure 6–5 shows the Stratix V transmitter in clock output mode. In clock output mode, you can use an LVDS channel as a clock output channel.

Figure 6–5. Stratix V Transmitter in Clock Output Mode



You can bypass the Stratix V serializer to support DDR (x2) and SDR (x1) operations to achieve a serialization factor of 2 and 1, respectively. The I/O element (IOE) contains two data output registers that can each operate in either DDR or SDR mode. Figure 6–6 shows the serializer bypass path.

Figure 6–6. Stratix V Serializer Bypass *(1), (2), (3)*



Notes to Figure 6–6:

(1) All disabled blocks and signals are grayed out.

(2) In DDR mode, `tx_inclock` clocks the IOE register. In SDR mode, data is directly passed through the IOE.

(3) In SDR and DDR modes, the data width to the IOE is 1 and 2 bits, respectively.

Programmable Differential Output Voltage (V_{OD}) and Programmable Pre-Emphasis

Stratix V LVDS transmitters support programmable pre-emphasis and programmable V_{OD}. Pre-emphasis increases the amplitude of the high-frequency component of the output signal, and thus helps to compensate for the frequency-dependent attenuation along the transmission line. Figure 6–7 shows the differential LVDS output.

Figure 6–7. Differential V_{OD}

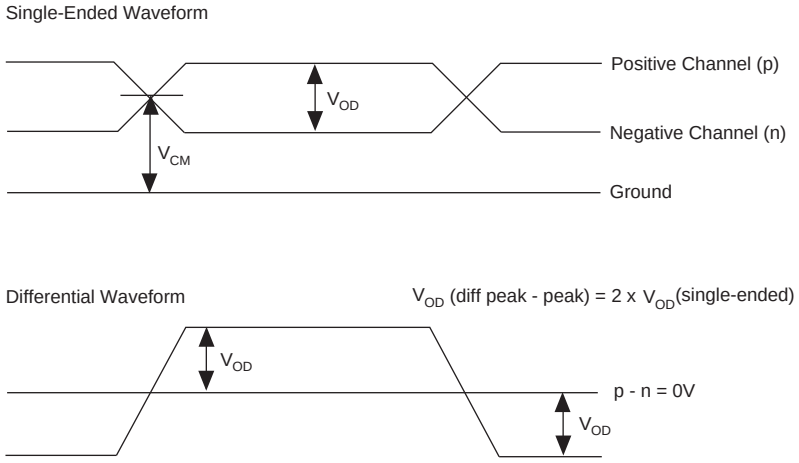
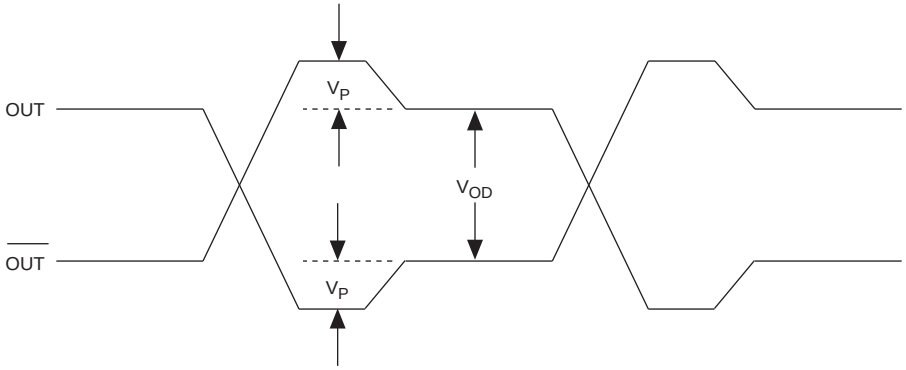


Figure 6–8 shows the LVDS output with pre-emphasis.

Figure 6–8. Programmable Pre-Emphasis *(1)*



Note to Figure 6–8:

(1) V_P— voltage boost from pre-emphasis. V_{OD}— differential output voltage (peak-peak).

Pre-emphasis is an important feature for high-speed transmission. Without pre-emphasis, the output current is limited by the V_{OD} setting and the output impedance of the driver. At high frequency, the slew rate may not be fast enough to reach full V_{OD} before the next edge, producing pattern-dependent jitter.

With pre-emphasis, the output current is boosted momentarily during switching to increase the output slew rate. The overshoot introduced by the extra current happens only during switching and does not ring, unlike the overshoot caused by signal reflection. The amount of pre-emphasis needed depends on the attenuation of the high-frequency component along the transmission line. The Quartus II software allows two settings for programmable pre-emphasis—disabled (**0**) and enabled (**1**). The default setting is enabled.

The V_{OD} is programmable with four settings: low (**0**), medium low (**1**), medium high (**2**), and high (**3**). The default setting is medium low.

Programmable V_{OD}

You can statically assign the V_{OD} settings from the Assignment Editor. Table 6–5 lists the assignment name for programmable V_{OD} and its possible values in the Quartus II software Assignment Editor.

Table 6–5. Quartus II Software Assignment Editor

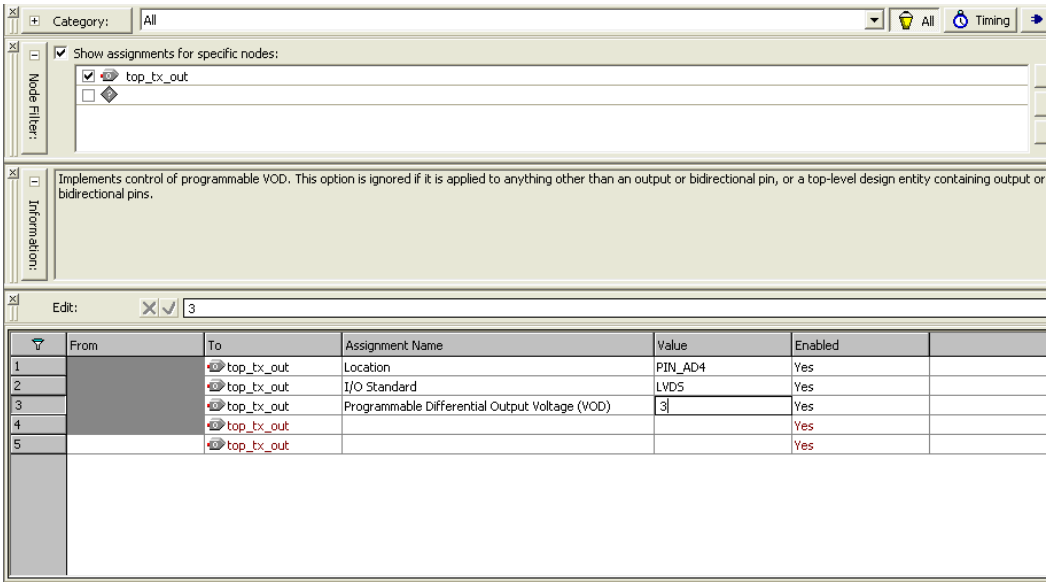
Field	Assignment
To	tx_out
Assignment name	Programmable Differential Output Voltage (V_{OD})
Allowed values (1)	0, 1, 2, 3

Note to Table 6–5:

(1) Value 0 is available for the **RSDS** I/O standard only and is not available for the **LVDS** I/O standard.

Figure 6–9 shows the assignment of programmable V_{OD} for a transmit data output from the Quartus II software Assignment Editor.

Figure 6–9. Quartus II Software Assignment Editor—Programmable V_{OD}



Programmable Pre-Emphasis

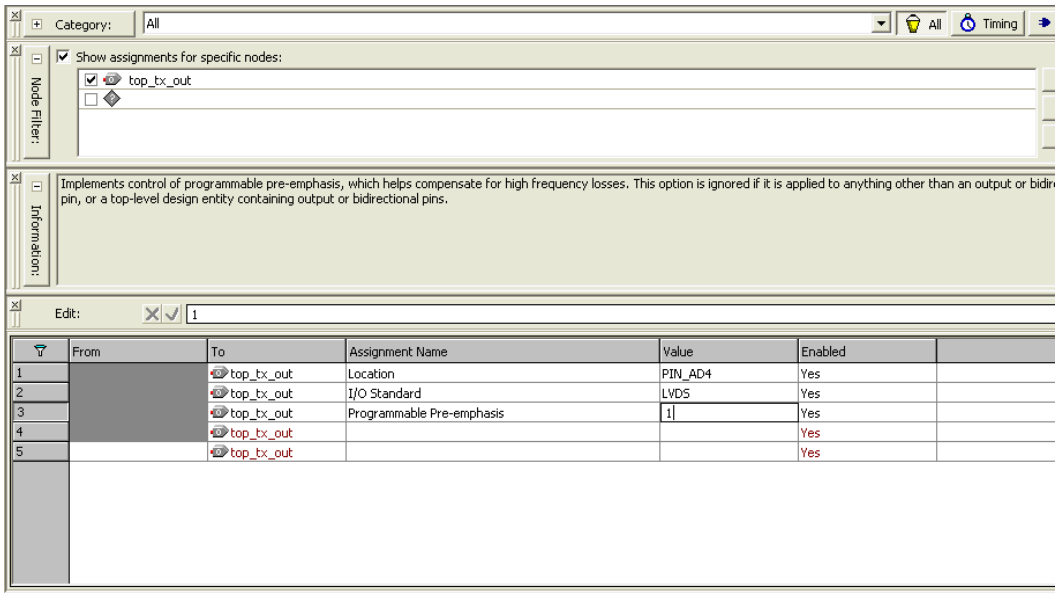
Two different settings are allowed for pre-emphasis from the Assignment Editor for each LVDS output channel. Table 6–6 lists the assignment name and its possible values for programmable pre-emphasis in the Quartus II software Assignment Editor.

Table 6–6. Quartus II Software Assignment Editor

Field	Assignment
To	`tx_out`
Assignment name	Programmable Pre-emphasis
Allowed values	0 (disable) and 1 (enable)

Figure 6–10 shows the assignment of programmable pre-emphasis for a transmit data output port from the Quartus II software Assignment Editor.

Figure 6–10. Quartus II Software Assignment Editor – Programmable Pre-Emphasis



Differential Receiver

Figure 6–11 shows the hardware blocks of the Stratix V receiver. The receiver has a differential buffer and fractional PLLs that can be shared between the transmitter and receiver, a DPA block, a synchronizer, a data realignment block, and a deserializer. The differential buffer can receive LVDS, mini-LVDS, and RSDS signal levels, which are statically set in the Quartus II software Assignment Editor.

For more information, refer to Figure 6–18 on page 6–20 and Figure 6–19 on page 6–21.

The fractional PLL receives the external clock input and generates different phases of the same clock. The DPA block chooses one of the clocks from the fractional PLL and aligns the incoming data on each channel. The synchronizer circuit is a 1-bit wide by 6-bit deep FIFO buffer that compensates for any phase difference between the DPA clock and the data realignment block. If necessary, the user-controlled data realignment circuitry inserts a single bit of latency in the serial bit stream to align to the word boundary. The deserializer includes shift registers and parallel load registers, and sends a maximum of 10 bits to the internal logic.

The Stratix V device family supports three different receiver modes:

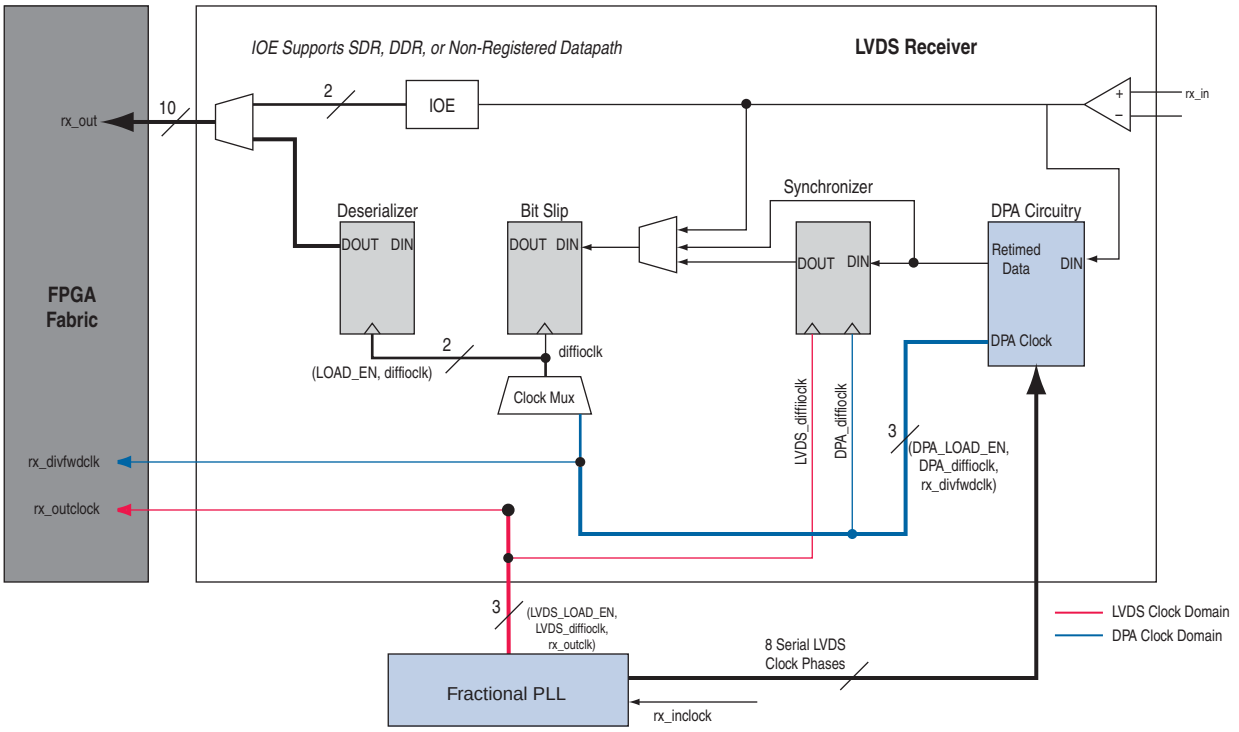
- "Non-DPA Mode" on page 6–18
- "DPA Mode" on page 6–20
- "Soft-CDR Mode" on page 6–21

The physical medium connecting the transmitter and receiver LVDS channels may introduce skew between the serial data and the source-synchronous clock. The instantaneous skew between each LVDS channel and the clock also varies with the jitter on the data and clock signals as seen by the receiver. The three different modes—non-DPA, DPA, and soft-CDR—provide different options to overcome skew between the source synchronous clock (non-DPA, DPA) /reference clock (soft-CDR) and the serial data.

Only the non-DPA mode requires manual skew adjustment.

Non-DPA mode allows you to statically select the optimal phase between the source synchronous clock and the received serial data to compensate skew. In DPA mode, the DPA circuitry automatically chooses the best phase to compensate for the skew between the source synchronous clock and the received serial data. Soft-CDR mode provides opportunities for synchronous and asynchronous applications for chip-to-chip and short reach board-to-board applications for SGMII protocols.

Figure 6–11. Receiver Block Diagram *(1), (2)*



Notes to Figure 6–11:

(1) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

(2) The `rx_out` port has a maximum data width of 10 bits.

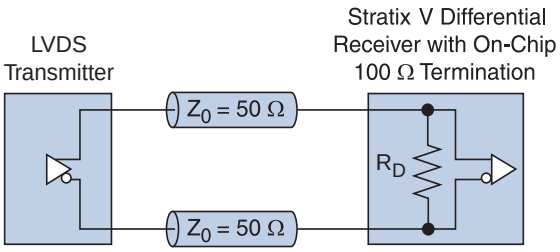
Differential I/O Termination

The Stratix V device family provides a 100-Ω, on-chip differential termination option on each differential receiver channel for LVDS standards. On-chip termination saves board space by eliminating the need to add external resistors on the board. You can enable on-chip termination in the Quartus II software Assignment Editor.

On-chip differential termination is supported on all I/O pins and dedicated clock input pins.

Figure 6–12 shows device on-chip termination.

Figure 6–12. On-Chip Differential I/O Termination



Receiver Hardware Blocks

The differential receiver has the following hardware blocks:

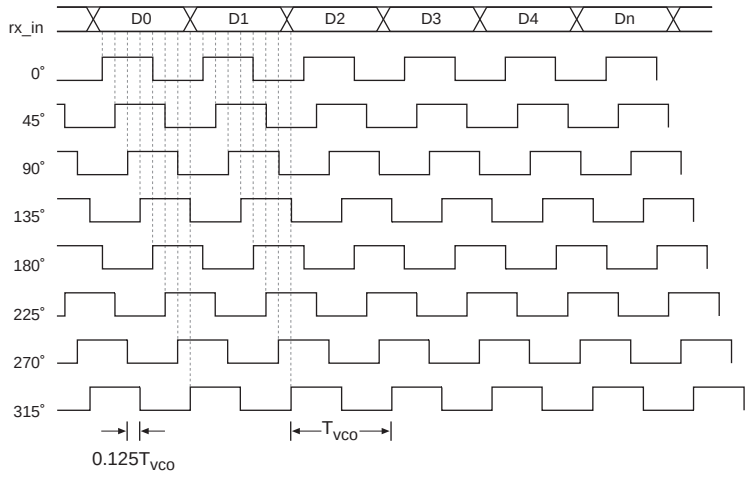
- "DPA Block" on page 6–14
- "Synchronizer" on page 6–15
- "Data Realignment Block (Bit Slip)" on page 6–15
- "Deserializer" on page 6–17

DPA Block

The DPA block takes in high-speed serial data from the differential input buffer and selects one of the eight phases generated by the fractional PLLs to sample the data. The DPA chooses a phase closest to the phase of the serial data. The maximum phase offset between the received data and the selected phase is 1/8 UI, which is the maximum quantization error of the DPA. The eight phases of the clock are equally divided, offering a 45° resolution.

Figure 6–13 shows the possible phase relationships between the DPA clocks and the incoming serial data.

Figure 6–13. DPA Clock Phase to Serial Data Timing Relationship *(1)*



Note to Figure 6–13:

(1) T_{VCO} is defined as the PLL serial clock period.

The DPA block continuously monitors the phase of the incoming serial data and selects a new clock phase if needed. You can prevent the DPA from selecting a new clock phase by asserting the optional RX_DPLL_HOLD port, which is available for each channel.

DPA circuitry does not require a fixed training pattern to lock to the optimum phase out of the eight phases. After reset or power up, DPA circuitry requires transitions on the received data to lock to the optimum phase. An optional output port, RX_DPA_LOCKED, is available to indicate an initial DPA lock condition to the optimum phase after power up or reset. This signal is not deasserted if the DPA selects a new phase out of the eight clock phases to sample the received data. Do not use the rx_dpa_locked signal to determine a DPA loss-of-lock condition. Use data checkers such as a cyclic redundancy check (CRC) or diagonal interleaved parity (DIP-4) to validate the data.

An independent reset port, RX_RESET, is available to reset the DPA circuitry. DPA circuitry must be retrained after reset.

The DPA block is bypassed in non-DPA mode.

Synchronizer

The synchronizer is a 1-bit wide and 6-bit deep FIFO buffer that compensates for the phase difference between DPA_diffioclk, which is the optimal clock selected by the DPA block, and LVDS_diffioclk, which is produced by the fractional PLLs. The synchronizer can only compensate for phase differences, not frequency differences between the data and the receiver's input reference clock.

An optional port, RX_FIFO_RESET, is available to the internal logic to reset the synchronizer. The synchronizer is automatically reset when the DPA first locks to the incoming data. Altera recommends using RX_FIFO_RESET to reset the synchronizer when the data checker indicates corrupted received data.

The synchronizer circuit is bypassed in non-DPA and soft-CDR mode.

Data Realignment Block (Bit Slip)

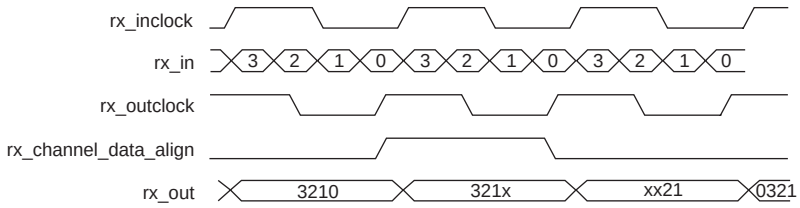
Skew in the transmitted data along with skew added by the link causes channel-to-channel skew on the received serial data streams. If the DPA is enabled, the received data is captured with different clock phases on each channel. This may cause the received data to be misaligned from channel to channel. To compensate for this channel-to-channel skew and establish the correct received word boundary at each channel, each receiver channel has a dedicated data realignment circuit that realigns the data by inserting bit latencies into the serial stream.

An optional RX_CHANNEL_DATA_ALIGN port controls the bit insertion of each receiver independently controlled from the internal logic. The data slips one bit on the rising edge of RX_CHANNEL_DATA_ALIGN. The requirements for the RX_CHANNEL_DATA_ALIGN signal include:

- The minimum pulse width is one period of the parallel clock in the logic array.
- The minimum low time between pulses is one period of the parallel clock.
- This is an edge-triggered signal.
- Valid data is available two parallel clock cycles after the rising edge of RX_CHANNEL_DATA_ALIGN.

Figure 6–14 shows receiver output (RX_OUT) after one bit slip pulse with the deserialization factor set to **4**.

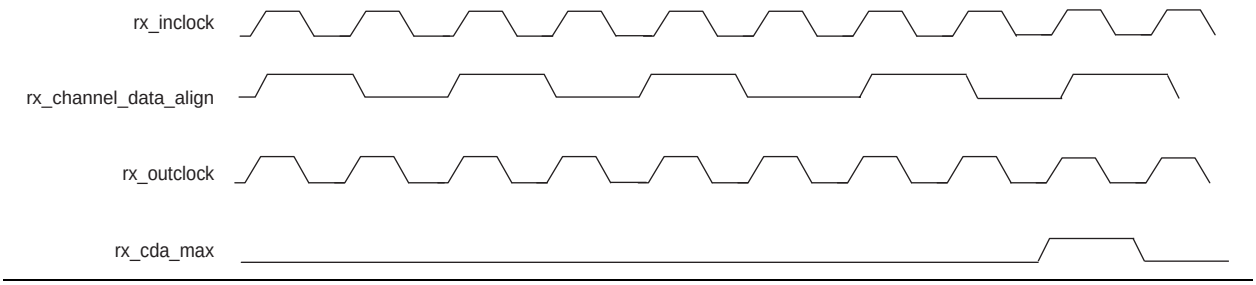
Figure 6–14. Data Realignment Timing



The data realignment circuit can have up to 11 bit-times of insertion before a rollover occurs. The programmable bit rollover point can be from 1 to 11 bit-times, independent of the deserialization factor. The programmable bit rollover point must be set equal to or greater than the deserialization factor, allowing enough depth in the word alignment circuit to slip through a full word. You can set the value of the bit rollover point using the MegaWizard Plug-In Manager. An optional status port, RX_CDA_MAX, is available to the FPGA fabric from each channel to indicate when the preset rollover point is reached.

Figure 6–15 shows a preset value of four bit-times before rollover occurs. The rx_cda_max signal pulses for one rx_outclock cycle to indicate that rollover has occurred.

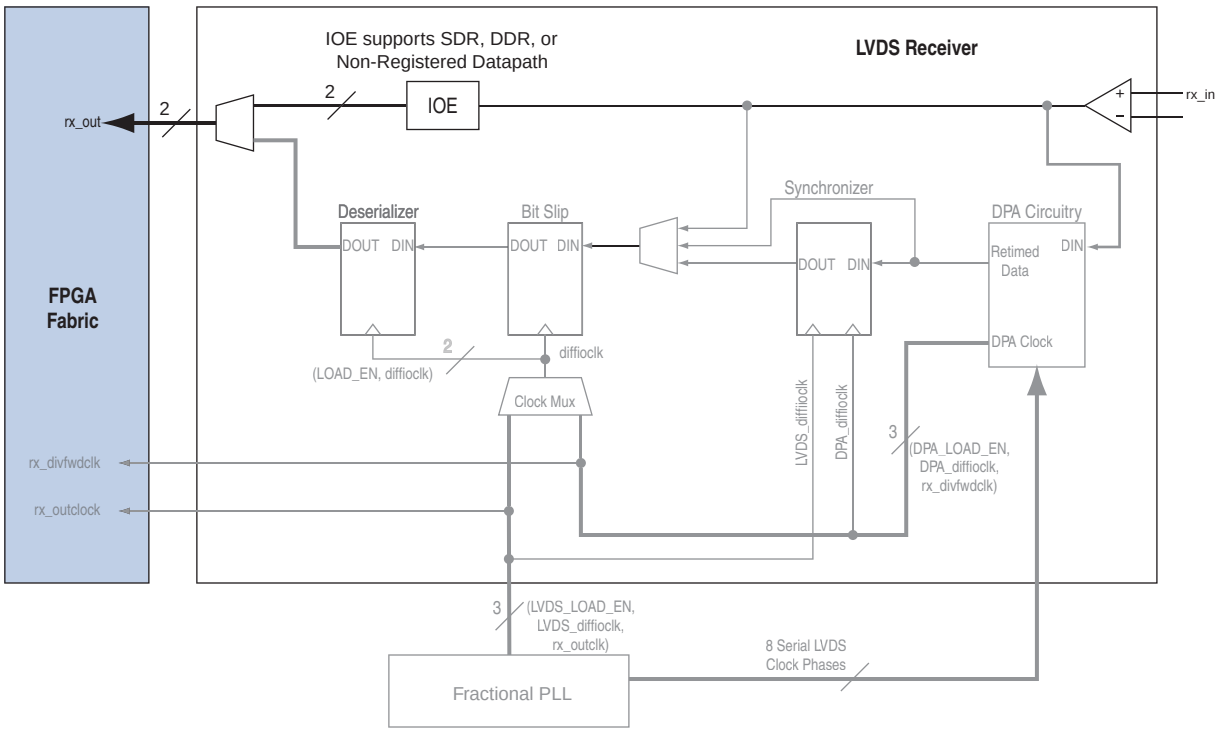
Figure 6–15. Receiver Data Realignment Rollover



Deserializer

You can statically set the deserialization factor to x3, x4, x5, x6, x7, x8, x9, or x10 by using the Quartus II software. You can bypass the Stratix V deserializer in the Quartus II MegaWizard Plug-In Manager to support DDR (x2) or SDR (x1) operations, as shown Figure 6–16. The DPA and data realignment circuit cannot be used when the deserializer is bypassed. The IOE contains two data input registers that can operate in DDR or SDR mode.

Figure 6–16. Stratix V Deserializer Bypass *(1), (2), (3)*



Notes to Figure 6–16:

(1) All disabled blocks and signals are grayed out.

(2) In DDR mode, `rx_inclock` clocks the IOE register. In SDR mode, data is directly passed through the IOE.

(3) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

Receiver Datapath Modes

The Stratix V device family supports three receiver datapath modes: non-DPA mode, DPA mode, and soft-CDR mode.

Non-DPA Mode

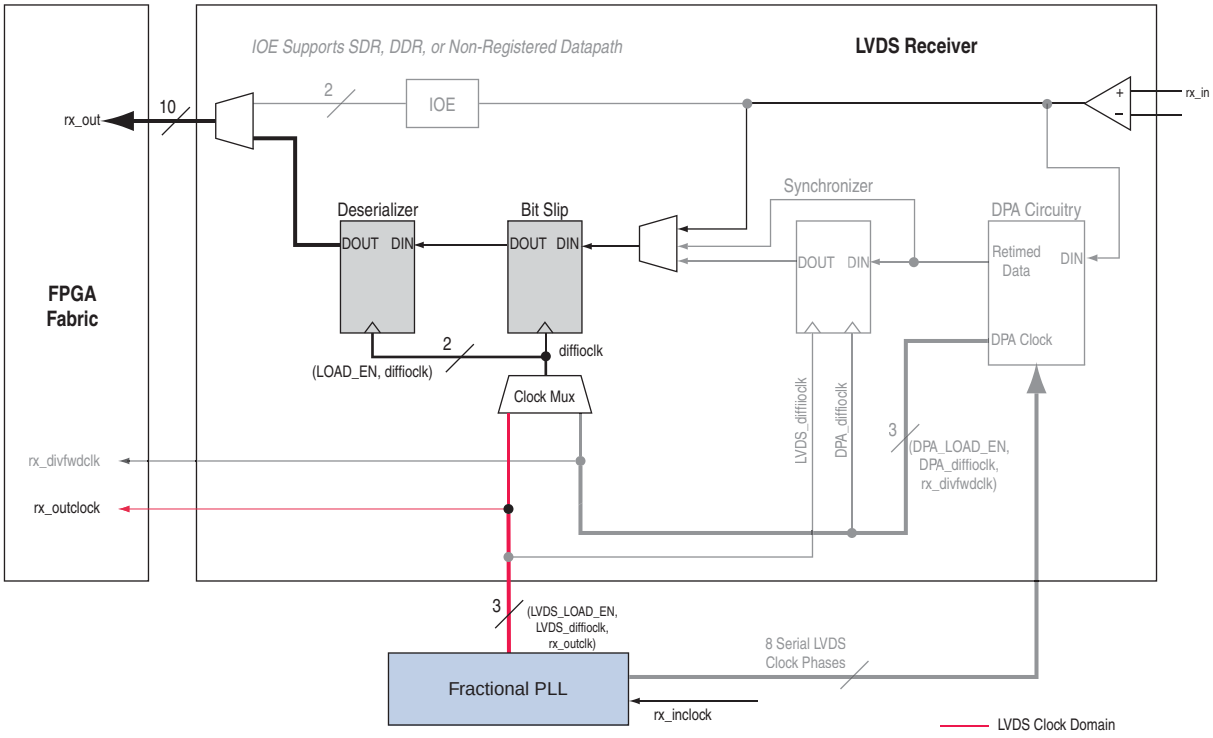
Figure 6–17 shows the non-DPA datapath block diagram. In non-DPA mode, the DPA and synchronizer blocks are disabled. Input serial data is registered at the rising or falling edge of the serial `LVDS_diffioclk` clock produced by the left and right PLLs. You can select the rising/falling edge option using the ALTLDVS MegaWizard Plug-In Manager. Both data realignment and deserializer blocks are clocked by the `LVDS_diffioclk` clock, which is generated by the left and right PLLs.

For Stratix V devices, you must perform PCB trace compensation to adjust the trace length of each LVDS channel to improve channel-to-channel skew when interfacing with non-DPA receivers at data rate above 840 Mbps.

The Quartus II software Fitter Report panel reports the amount of delay you must add to each trace for the Stratix V device. You can use the recommended trace delay numbers published under the LVDS Transmitter/Receiver Package Skew Compensation panel and manually compensate the skew on the PCB board trace to reduce channel-to-channel skew, thus meeting the timing budget between LVDS channels.

For more information about the LVDS Transmitter/Receiver Package Skew Compensation report panel, refer to the "Arria II GX and Stratix V LVDS Package Skew Compensation Report Panel" section in the *LVDS SERDES Transmitter/Receiver (ALTLVDS_TX and ALTLVDS_RX) Megafunction User Guide*.

Figure 6–17. Receiver Datapath in Non-DPA Mode *(1), (2), (3)*



Notes to Figure 6–17:

(1) All disabled blocks and signals are grayed out.

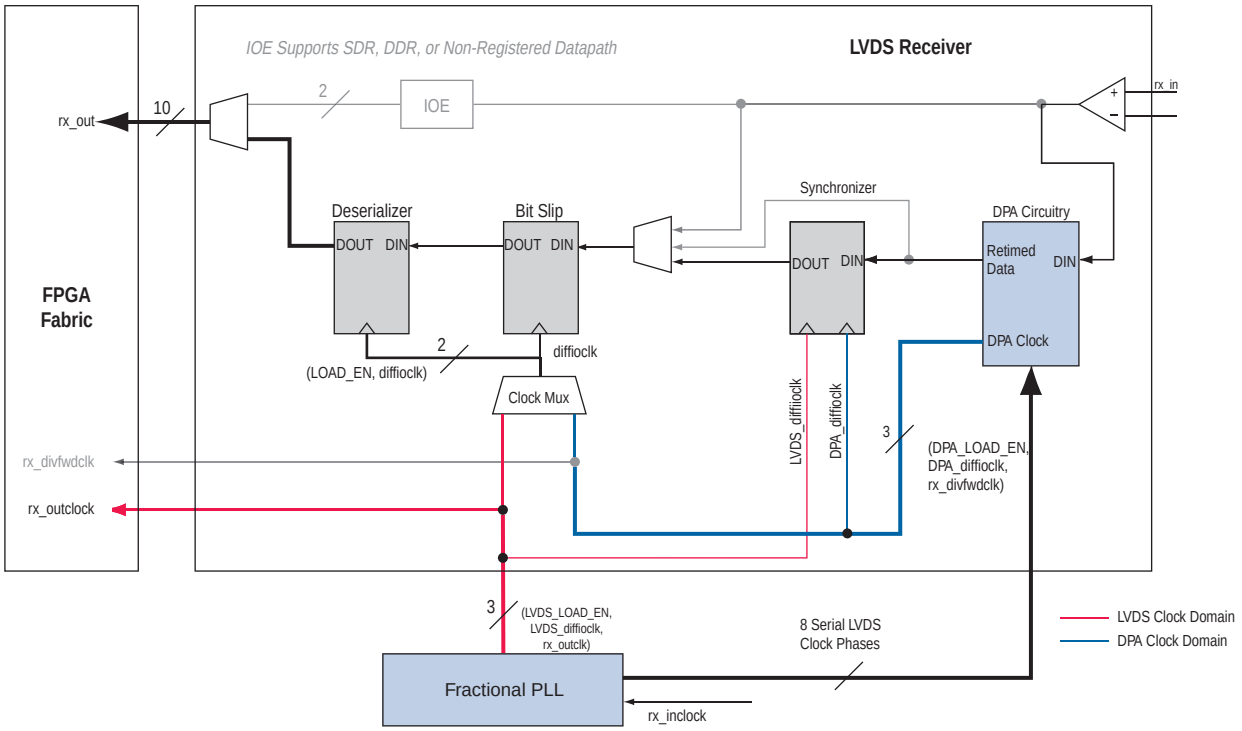
(2) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

(3) The `rx_out` port has a maximum data width of 10 bits.

DPA Mode

Figure 6–18 shows the DPA mode datapath, where all the hardware blocks mentioned in "Receiver Hardware Blocks" on page 6–14 are active. The DPA block chooses the best possible clock (DPA_diffioclk) from the eight fast clocks sent by the fractional PLL. This serial DPA_diffioclk clock is used for writing the serial data into the synchronizer. A serial LVDS_diffioclk clock is used for reading the serial data from the synchronizer. The same LVDS_diffioclk clock is used in data realignment and deserializer blocks.

Figure 6–18. Receiver Datapath in DPA Mode *(1), (2), (3)*



Notes to Figure 6–18:

(1) All disabled blocks and signals are grayed out.

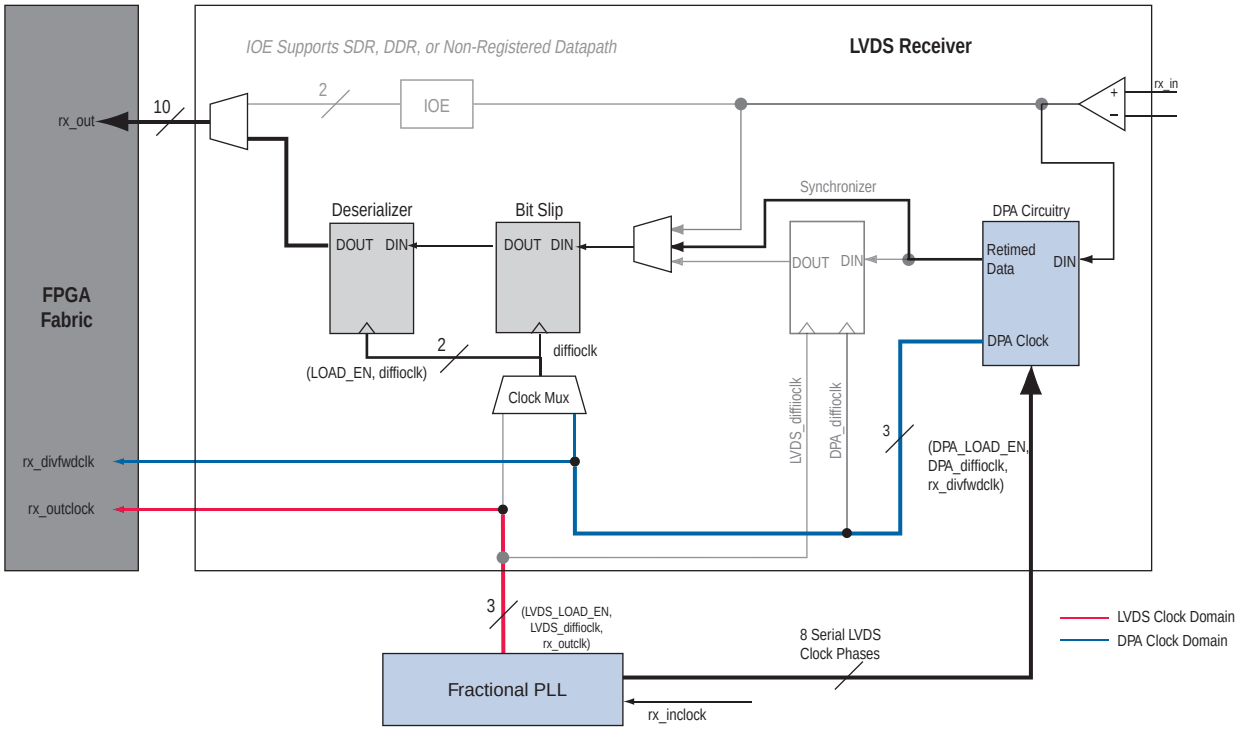
(2) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

(3) The rx_out port has a maximum data width of 10 bits.

Soft-CDR Mode

The Stratix V LVDS channel offers the soft-CDR mode to support the GbE and SGMII protocols. A receiver PLL uses the local clock source for reference. Figure 6–19 shows the soft-CDR mode datapath.

Figure 6–19. Receiver Datapath in Soft-CDR Mode *(1), (2), (3)*



Notes to Figure 6–19:

(1) All disabled blocks and signals are grayed out.
(2) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.
(3) The `rx_out` port has a maximum data width of 10 bits.

In soft-CDR mode, the synchronizer block is inactive. The DPA circuitry selects an optimal DPA clock phase to sample the data. Use the selected DPA clock for bit-slip operation and deserialization. The DPA block also forwards the selected DPA clock, divided by the deserialization factor called `rx_divfwdclk`, to the FPGA fabric, along with the deserialized data. This clock signal is put on the periphery clock (PCLK) network. When using soft-CDR mode, the `rx_reset` port must not be asserted after DPA training because the DPA will continuously choose new phase taps from the PLL to track parts per million (PPM) differences between the reference clock and incoming data.

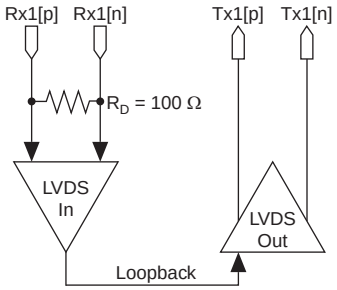
For more information about PCLK networks, refer to the *Clock Networks and PLLs in Stratix V Devices* chapter.

You can use every LVDS channel in soft-CDR mode and drive the FPGA fabric using the PCLK network in the Stratix V device family. The `rx_dpa_locked` signal is not valid in soft-CDR mode because the DPA continuously changes its phase to track PPM differences between the upstream transmitter and the local receiver input reference clocks. The parallel clock `rx_outclock`, generated by the left and right PLLs, is also forwarded to the FPGA fabric.

LVDS Direct Loopback Mode

The Stratix V device family supports direct loopback mode for the LVDS driver and receiver pairs within the same LVDS module only. An LVDS module is a pair of receiver and transmitter pins which share the same logic array block (LAB) row or LAB column. Figure 6–20 shows the true LVDS input and output buffer from an I/O pair from the same module. LVDS direct loopback mode allows you to verify the LVDS driver and receiver pair by checking the incoming LVDS data from the true LVDS input buffer into the true LVDS output buffer.

Figure 6–20. LVDS Direct Loopback Path *(1)*



Note to Figure 6–20:

(1) The R_D value is pending characterization.

The Quartus II software allows two option settings for LVDS direct loopback mode in the assignment editor: **On** and **Off**.

This option is only available for true differential I/O standards only.

This option can be applied on the LVDS output pair that is already being used in the design. Turning the LVDS direct loopback mode option to **On** overrides the connection from core with the signal from the true differential input buffer in the same I/O module. You can disable this option after verifying the LVDS driver receiver pair. Turn the LVDS direct loopback mode option to **Off** and recompile your design.

LVDS Interface with the Use External PLL Option Enabled

The ALTLVDS MegaWizard Plug-In Manager provides an option for implementing the LVDS interface with the **Use External PLL** option. With this option enabled you can control the PLL settings, such as dynamically reconfiguring the PLL to support different data rates, dynamic phase shift, and other settings. You also must instantiate an Altera_PLL megafunction to generate the various clock and load enable signals.

When you enable the **Use External PLL** option with the ALTLVDS transmitter and receiver, the following signals are required from the Altera_PLL megafunction:

- Serial clock input to the SERDES of the ALTLVDS transmitter and receiver
- Load enable to the SERDES of the ALTLVDS transmitter and receiver
- Parallel clock used to clock the transmitter FPGA fabric logic and parallel clock used for the receiver `rx_syncclock` port and receiver FPGA fabric logic
- Asynchronous PLL reset port of the ALTLVDS receiver

As an example, Table 6–7 lists the serial clock output, load enable output, and parallel clock output generated on ports c0, c1, and c2, respectively, along with the locked signal of the Altera_PLL instance. You can choose any of the PLL output clock ports to generate the interface clocks.

With soft SERDES, a different clocking requirement is needed. For more information, refer to the *LVDS SERDES Transmitter/Receiver (ALTLVDS_RX/TX) Megafunction User Guide*.

The high-speed clock generated from the PLL is intended to clock the LVDS SERDES circuitry only. Do not use the high-speed clock to drive other logic because the allowed frequency to drive the core logic is restricted by the PLL F_{OUT} specification. For more information about the F_{OUT} specification, refer to the *Stratix V Device Datasheet*.

Table 6–7 lists the signal interface between the output ports of the Altera_PLL megafunction and the input ports of the ALTLVDS transmitter and receiver.

Table 6–7. Signal Interface Between Altera_PLL and ALTLVDS Megafunctions (Part 1 of 2)

From the Altera_PLL Megafunction	To the ALTLVDS Transmitter	To the ALTLVDS Receiver
Serial clock output (c0) *(1)*	`tx_inclock` (serial clock input to the transmitter)	`rx_inclock` (serial clock input) `rx_dpaclock`
Load enable output (c1)	`tx_enable` (load enable to the transmitter)	`rx_enable` (load enable for the deserializer)
Parallel clock output (c2)	Parallel clock used inside the transmitter core logic in the FPGA fabric	`rx_syncclock` (parallel clock input) and parallel clock used inside the receiver core logic in the FPGA fabric

Table 6–7. Signal Interface Between Altera_PLL and ALTLVDS Megafunctions (Part 2 of 2)

From the Altera_PLL Megafunction	To the ALTLVDS Transmitter	To the ALTLVDS Receiver
~(locked)	—	pll_areset (asynchronous PLL reset port) (2)

Notes to Table 6–7:

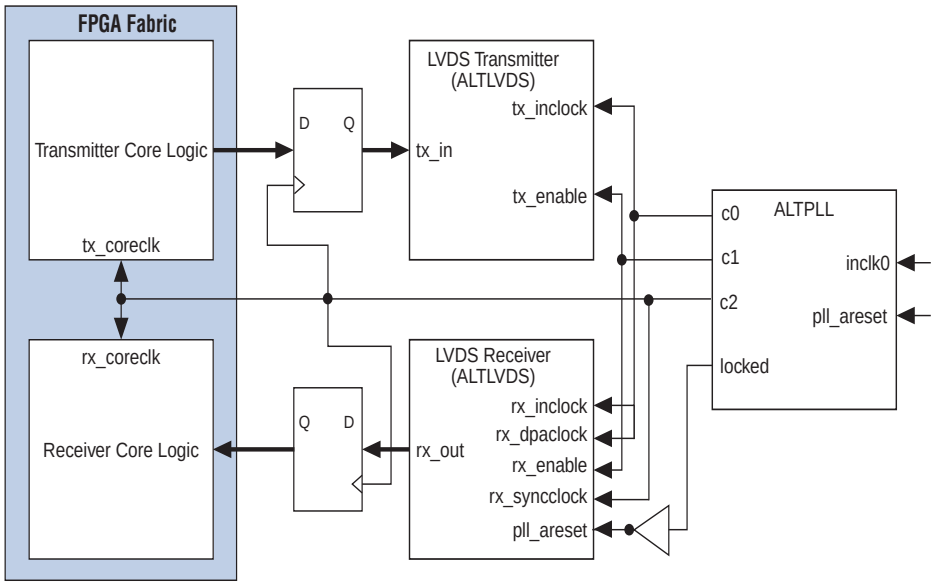
(1) The serial clock output (c0) can only drive tx_inclock on the ALTLVDS transmitter and rx_inclock and rx_dpaclock on the ALTLVDS receiver. This clock cannot drive the core logic.

(2) The pll_areset signal is automatically enabled for the LVDS receiver in external PLL mode. This signal does not exist for LVDS transmitter instantiation when the external PLL option is enabled.

The rx_syncclock port is automatically enabled in an LVDS receiver in external PLL mode. The Quartus II compiler errors out if this port is not connected, as shown in Figure 6–21.

When generating the Altera_PLL megafunction, the **Left/Right PLL** option is configured to set up the PLL in LVDS mode. Figure 6–21 shows the connection between the Altera_PLL and ALTLVDS megafunctions.

Figure 6–21. LVDS Interface with the Altera_PLL Megafunction (1)



Note to Figure 6–21:

(1) Instantiation of pll_areset is optional for the Altera_PLL instantiation.

Example 6–1 shows how to generate three output clocks using an Altera_PLL megafunction.

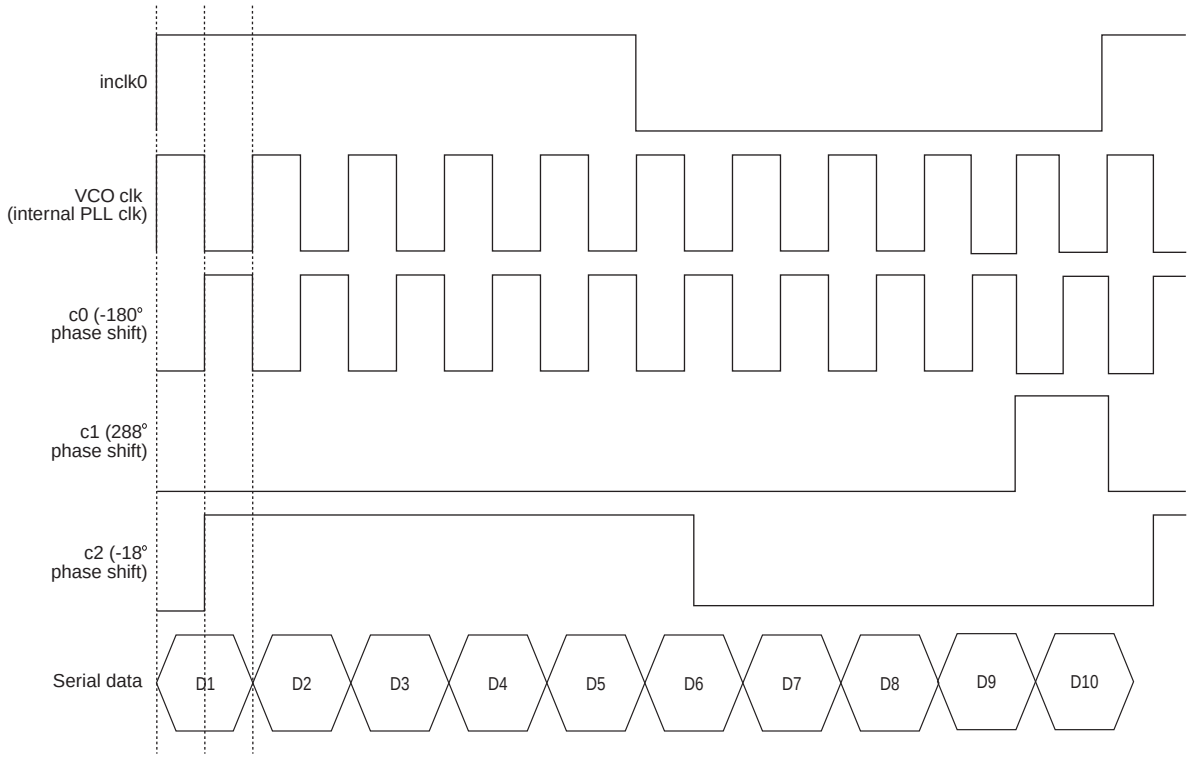
Example 6–1. Generating Three Output Clocks Using an Altera_PLL Megafunction

Altera_PLL clocking requirements for ALTLVDS_TX and ALTLVDS_RX when not using DPA and soft-CDR mode:

- c0
  - Frequency = data rate
  - Phase shift = –180 + 360°
  - Duty cycle = 50%
  - Connects to the tx_inclock port of ALTLVDS_TX and the rx_inclock port of ALTLVDS_RX
- c1
  - Frequency = data rate/serialization factor
  - Phase shift = [(serialization factor - 2)/serialization factor] × 360°
  - Duty cycle = 100/serialization factor
  - Connects to the tx_enable port of ALTLVDS_TX and the rx_enable port of ALTLVDS_RX
- c2
  - Frequency = data rate/serialization factor
  - Phase shift = [(–180/serialization factor) + 360°]
  - Duty cycle = 50%
  - Used as the core clock for the parallel data registers for both tx and rx and connects to the rx_synclock port of ALTLVDS_RX

The Equation 6–1 calculations for phase shift assume that the input clock and serial data are edge aligned. Introducing a phase shift of –180° to sampling clock (c0) ensures that the input data is center-aligned with respect to the c0, as shown in Figure 6–22.

Figure 6–22. Phase Relationship for External PLL Interface Signals

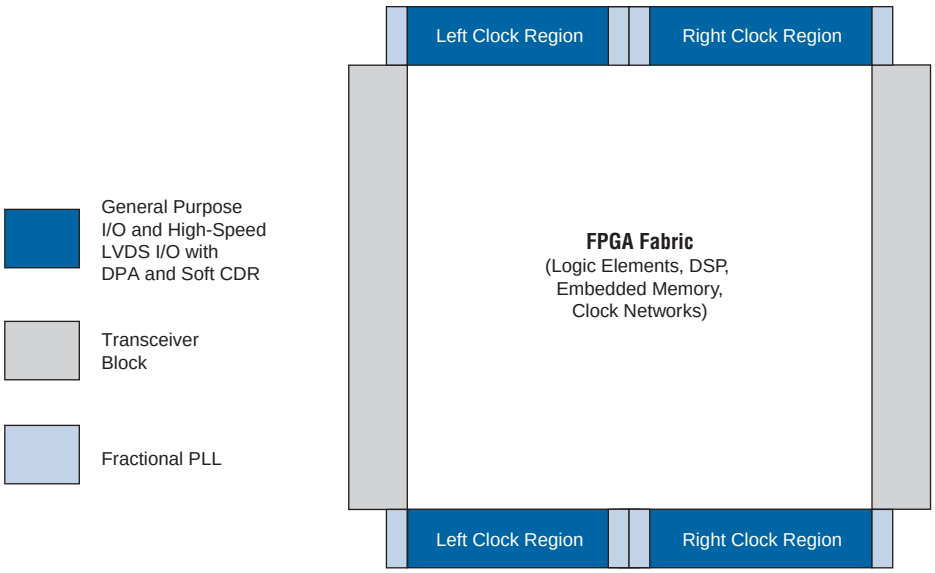


PLLs and Stratix V Clocking

The Stratix V device family supports fractional PLLs on each side of the device. Figure 6–23 shows the location of the fractional PLLs supported for the high-speed differential I/O receiver and transmitter channels to generate the parallel clocks (`rx_outclock` and `tx_outclock`) and high-speed clocks (`diffioclk`).

For more information about fractional PLLs, refer to the *Clock Network and PLLs in Stratix V Devices* chapter.

Figure 6–23. Fractional PLL Locations for High-Speed Differential I/Os with DPA Locations Stratix V Devices


Left Clock Region
Right Clock Region
General Purpose I/O and High-Speed LVDS I/O with DPA and Soft CDR
Transceiver Block
Fractional PLL
FPGA Fabric
(Logic Elements, DSP, Embedded Memory, Clock Networks)
Left Clock Region
Right Clock Region

To drive the LVDS channels, you must use the PLLs in integer PLL mode. The center or corner PLLs can drive the LVDS receiver and driver channels. However, the clock tree network cannot cross over to different I/O regions. For example, the top left corner PLL cannot cross over to drive the LVDS receiver and driver channels on the top right I/O bank when the top center PLL is driving any of the LVDS receiver and driver channels on the top left I/O bank.

For more information on the fractional PLL clocking restrictions, refer to the "Differential Pin Placement Guidelines" on page 6–35.

Source-Synchronous Timing Budget

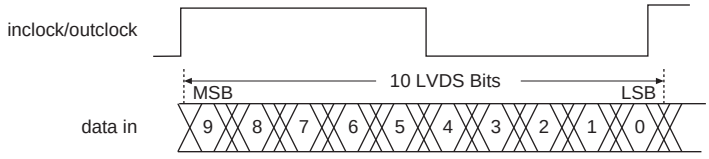
This section describes the timing budget, waveforms, and specifications for source-synchronous signaling in the Stratix V device family. LVDS I/O standards enable high-speed data transmission. This high data transmission rate results in better overall system performance. To take advantage of fast system performance, you must understand how to analyze timing for these high-speed signals. Timing analysis for the differential block is different from traditional synchronous timing analysis techniques.

Instead of focusing on clock-to-output and setup times, source synchronous timing analysis is based on the skew between the data and the clock signals. High-speed differential data transmission requires the use of timing parameters provided by IC vendors and is strongly influenced by board skew, cable skew, and clock jitter. This section defines the source-synchronous differential data orientation timing parameters, the timing budget definitions for the Stratix V device family, and how to use these timing parameters to determine a design's maximum performance.

Differential Data Orientation

There is a set relationship between an external clock and the incoming data. For operations at 1 Gbps and a serialization factor of 10, the external clock is multiplied by 10. You can set phase-alignment in the PLL to coincide with the sampling window of each data bit. The data is sampled on the falling edge of the multiplied clock. Figure 6–24 shows the data bit orientation of the x10 mode.

Figure 6–24. Bit Orientation in the Quartus II Software



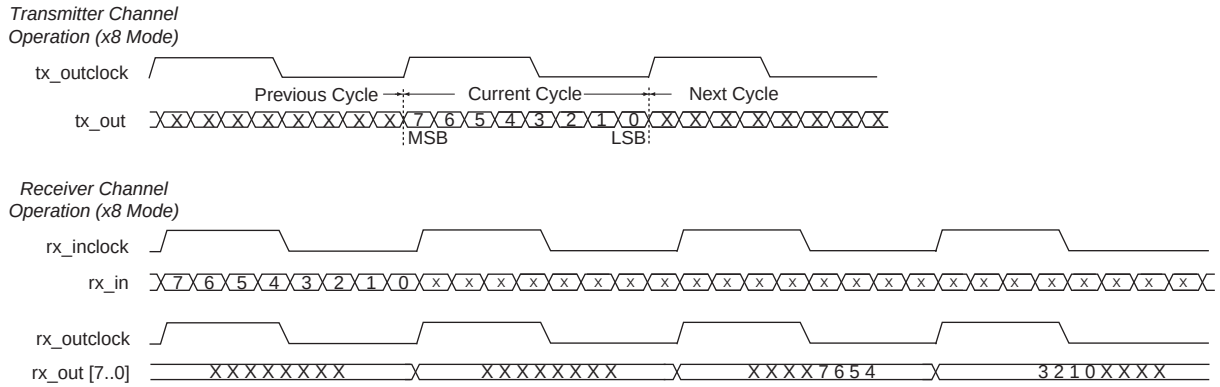
Differential I/O Bit Position

Data synchronization is necessary for successful data transmission at high frequencies. Figure 6–25 shows the data bit orientation for a channel operation. This figure is based on the following:

- Serialization factor equals the clock multiplication factor
- Edge alignment is selected for phase alignment
- Implemented in hard SERDES

For other serialization factors, use the Quartus II software tools to find the bit position within the word. The bit positions after deserialization are listed in Table 6–8.

Figure 6–25. Bit-Order and Word Boundary for One Differential Channel *(1)*



Note to Figure 6–25:

(1) These are only functional waveforms and are not intended to convey timing information.

Table 6–8 lists the conventions for differential bit naming for 18 differential channels. The MSB and LSB positions increase with the number of channels used in a system.

Table 6–8. Differential Bit Naming

Receiver Channel Data Number	Internal 8-Bit Parallel Data	
	MSB Position	LSB Position
1	7	0
2	15	8
3	23	16
4	31	24
5	39	32
6	47	40
7	55	48
8	63	56
9	71	64
10	79	72
11	87	80
12	95	88
13	103	96
14	111	104
15	119	112
16	127	120
17	135	128
18	143	136

Transmitter Channel-to-Channel Skew

Transmitter channel-to-channel skew (TCCS) is an important parameter based on the Stratix V transmitter in a source synchronous differential interface. This parameter is used in receiver skew margin calculation. For more information, refer to "Receiver Skew Margin for Non-DPA Mode" on page 6–30.

TCCS is the difference between the fastest and slowest data output transitions, including the TCO variation and clock skew. For LVDS transmitters, the TimeQuest Timing Analyzer provides a TCCS report, which shows TCCS values for serial output ports.

You can get the TCCS value from the TCCS report (`report_TCCS`) in the Quartus II compilation report in the TimeQuest Timing Analyzer, or from the *Stratix V Device Datasheet*.

Receiver Skew Margin for Non-DPA Mode

Changes in system environment, such as temperature, media (cable, connector, or PCB), and loading affect the receiver's setup and hold times; internal skew affects the sampling ability of the receiver.

Different modes of LVDS receivers use different specifications that can help in deciding the ability to sample the received serial data correctly. In DPA mode, you must use DPA jitter tolerance instead of receiver input skew margin (RSKM).

In non-DPA mode, use TCCS, RSKM, and sampling window (SW) specifications for high-speed source-synchronous differential signals in the receiver datapath. The relationship between RSKM, TCCS, and SW is expressed by the RSKM equation shown in Equation 6–1.

Equation 6–1. RSKM

$$RSKM = \frac{TUI - SW - TCCS}{2}$$

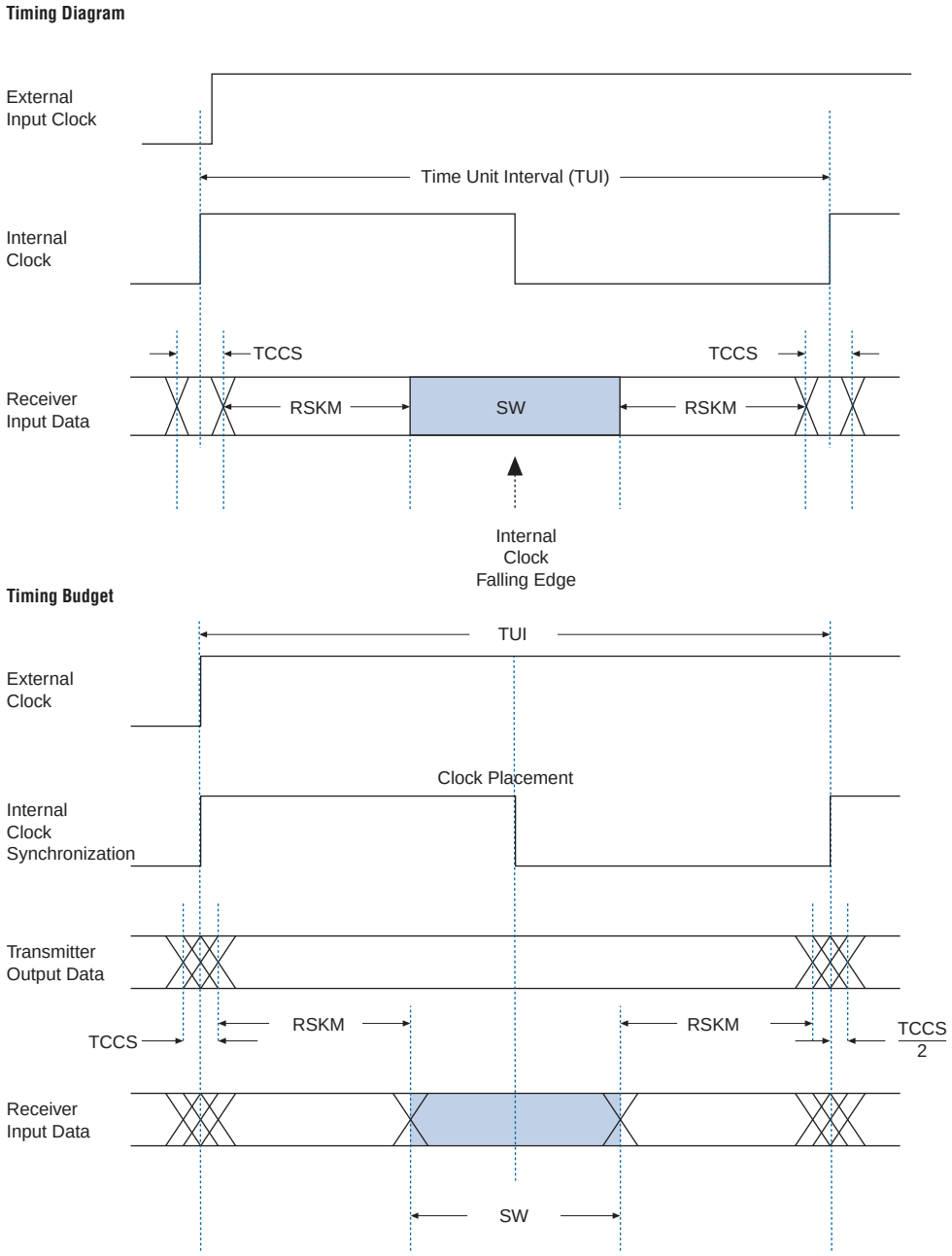
Conventions used for the equation:

- RSKM—the timing margin between the receiver's clock input and the data input sampling window.
- Time unit interval (TUI)—time period of the serial data.
- SW—the period of time that the input data must be stable to ensure that data is successfully sampled by the LVDS receiver. The SW is a device property and varies with device speed grade.
- TCCS—the timing difference between the fastest and the slowest output edges, including t_{CO} variation and clock skew, across channels driven by the same PLL. The clock is included in the TCCS measurement.

Figure 6–26 shows the relationship between the RSKM, TCCS, and the receiver's SW.

You must calculate the RSKM value to decide whether or not data can be sampled properly by the LVDS receiver with the given data rate and device. A positive RSKM value indicates that the LVDS receiver can sample the data properly, whereas a negative RSKM indicates that it cannot.

Figure 6–26. Differential High-Speed Timing Diagram and Timing Budget for Non-DPA Mode



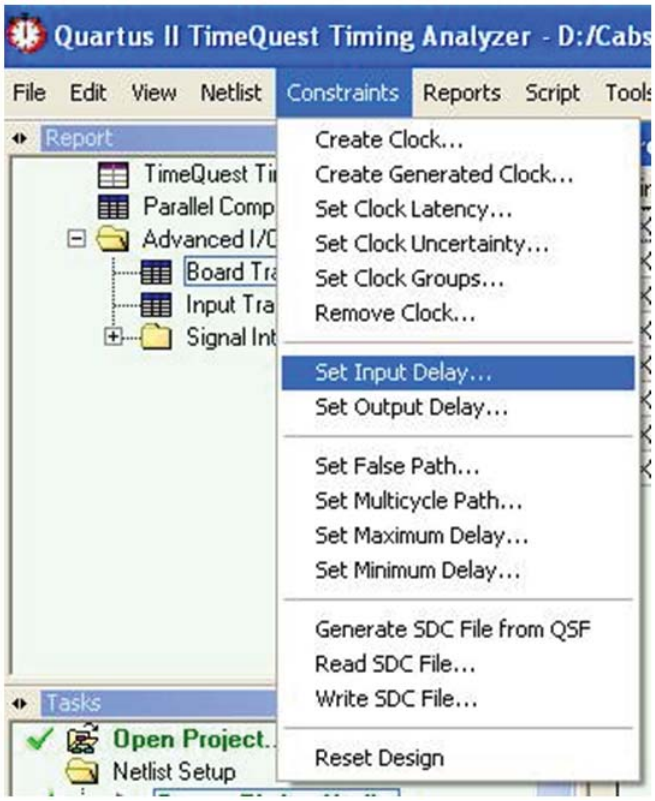
For LVDS receivers, the Quartus II software provides an RSKM report showing the SW, TUI, and RSKM values for non-DPA mode. You can generate the RSKM report by executing the `report_RSKM` command in the TimeQuest Timing Analyzer. You can find the RSKM report in the Quartus II compilation report in the TimeQuest Timing Analyzer section.

To obtain the RSKM value, assign an appropriate input delay to the LVDS receiver through the TimeQuest Timing Analyzer constraints menu.

For assigning an input delay, follow these steps:

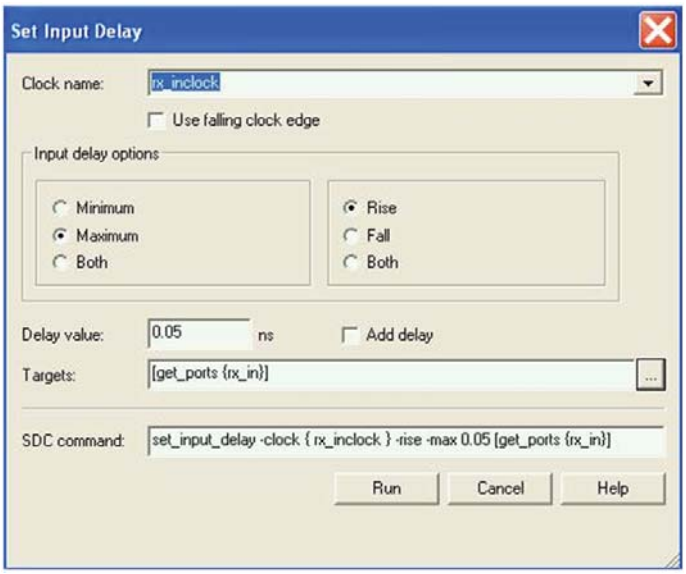
1. The Quartus II TimeQuest Timing Analyzer GUI has many options for setting the constraints and analyzing the design. Figure 6–27 shows various commands on the Constraints menu. For setting an input delay, you must select the **Set Input Delay** option.

Figure 6–27. Selection of Constraints Menu in TimeQuest Timing Analyzer



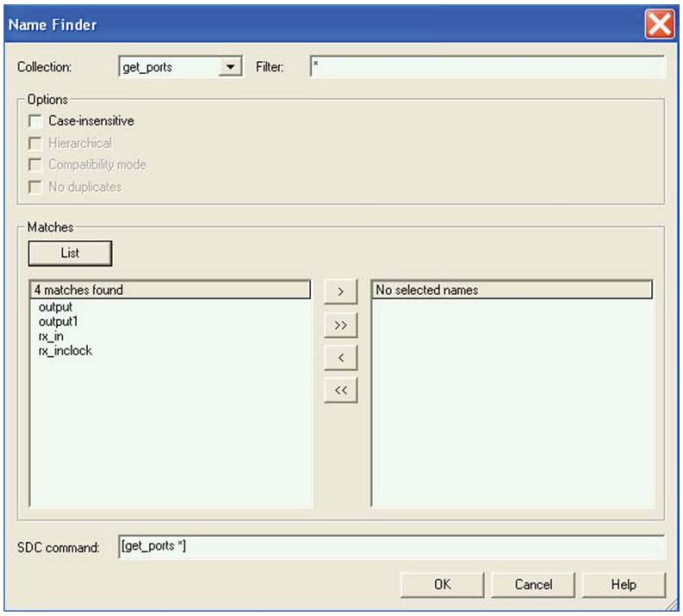
2. Figure 6–28 shows the setting parameters for the **Set Input Delay** option. The clock name must reference the source synchronous clock that feeds the LVDS receiver. Select the desired clock using the pull-down menu.

Figure 6–28. Input Time Delay Assignment Through TimeQuest Timing Analyzer



3. Click the **Browse** button to the right of the **Targets** option. You can view a list of all available ports using the **List** option in the **Name Finder** window (Figure 6–29).

Figure 6–29. Name Finder Window in Set Input Delay Option



4. Select the LVDS receiver serial input ports (from the list) according to the input delay you set. Click **OK**.
5. In the **Set Input Delay** window, set the appropriate values in the **Input Delay Options** section and **Delay value**.
6. Click **Run** to incorporate these values in the TimeQuest Timing Analyzer.
7. Assign the appropriate delay for all the LVDS receiver input ports following these steps. If you have already assigned **Input Delay** and you need to add more delay to that input port, use the **Add Delay** option in the **Set Input Delay** window.

If no input delay is set in the TimeQuest Timing Analyzer, the receiver channel-to-channel skew (RCCS) defaults to zero. You can also directly set the input delay in a synopsys design constraint file (**.sdc)** using the `set_input_delay` command.

For more information about **.sdc** commands and the TimeQuest Timing Analyzer, refer to the *Quartus II TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Development Software Handbook*.

Example 6–2 shows the RSKM calculation.

Example 6–2. RSKM

Data Rate: 1 Gbps, Board channel-to-channel skew = **200 ps**

For Stratix V devices:

TCCS = **100 ps** (pending characterization)
SW = **300 ps** (pending characterization)

TUI = **1000 ps**
Total RCCS = TCCS + Board channel-to-channel skew= 100 ps + 200 ps
= **300 ps**

RSKM= (**TUI - SW - RCCS)/2**
= (**1000 ps — 300 ps — 300 ps)/2**
= **200 ps**

```
Because the RSKM > 0 ps, receiver non-DPA mode must work correctly.
```

You can also calculate RSKM using the steps described in "Guidelines for DPA-Enabled and DPA-Disabled Differential Channels" on page 6–35.

Differential Pin Placement Guidelines

To ensure proper high-speed operation, differential pin placement guidelines have been established. The Quartus II compiler automatically checks that these guidelines are followed and issues an error message if they are not met. This section is divided into pin placement guidelines with and without DPA usage because DPA usage adds some constraints on the placement of high-speed differential channels.

☞ DPA-enabled differential channels refer to DPA mode or soft-CDR mode; DPA disabled channels refer to non-DPA mode.

☞ The information in the following sections is preliminary.

Guidelines for DPA-Enabled and DPA-Disabled Differential Channels

The Stratix V device family has differential receivers and transmitters in all I/O blocks. Each receiver has a dedicated DPA circuit to align the phase of the clock to the data phase of its associated channel. When you use DPA-enabled channels in differential banks, you must adhere to the guidelines listed in the following sections.

DPA-Enabled Channels, DPA-Disabled Channels, and Single-Ended I/Os

When you enable a DPA channel in a bank, both single-ended I/Os and differential I/O standards are allowed in the bank.

Double data rate I/O (DDIO) output pins can be placed within I/O modules that have the same pad group number as a SERDES differential channel but half rate DDIO (single data rate) output pins cannot be placed within I/O modules that have the same pad group number as a receiver SERDES differential channel. The input register must be implemented within the FPGA fabric logic.

The following lists two DPA restrictions:

- Because there is only a single DPA clock bus, a PLL drives a continuous series of DPA channels.
- To prevent noise mixing, use one row of separation between two groups of DPA channels. The two PLLs operate using different reference clocks (either same base frequency with PPM differences or an entirely different frequencies).

Document Revision History

Table 6–9 lists the revision history for this chapter.

Table 6–9. Document Revision History

Date	Version	Changes
June 2012	1.4	■ Added Table 6–2. ■ Updated Table 6–1, Table 6–3, Table 6–4, and Table 6–5. ■ Updated Figure 6–22. ■ Updated "Non-DPA Mode", "Soft-CDR Mode", and "PLLs and Stratix V Clocking" sections.
November 2011	1.3	■ Updated Table 6–2. ■ Updated Example 6–1. ■ Updated "LVDS Direct Loopback Mode" and "LVDS Interface with the Use External PLL Option Enabled" sections.
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Added Table 6–2 and Table 6–3. ■ Updated Table 6–1. ■ Updated Figure 6–2 and Figure 6–23. ■ Updated "Locations of the I/O Banks", "Programmable Pre-Emphasis", "Differential Receiver", "Fractional PLLs and Stratix V Clocking", and "DPA-Enabled Channels, DPA-Disabled Channels, and Single-Ended I/Os" sections. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

7. External Memory Interfaces in Stratix V Devices

SV51008-1.4

This chapter describes external memory interfaces available with Stratix® V devices, as well as the silicon capabilities of Stratix V devices to support external memory interfaces. Stratix V devices provide an efficient architecture to quickly and easily fit wide external memory interfaces to support the high level of system bandwidth within their small modular I/O bank structure. The I/Os are designed to provide high-performance support for existing and emerging external double data rate (DDR) memory standards, such as DDR3 and DDR2 SDRAM, QDR II+ and QDR II SRAM, and RLDRAM II.

Stratix V I/O elements (IOEs) provide easy-to-use built-in functionality required for a rapid and robust implementation with features such as dynamic calibrated on-chip termination (OCT), trace mismatch compensation, read- and write-leveling support for DDR3 SDRAM interfaces, read FIFO blocks, and 4- to 36-bit programmable DQ group widths.

The high-performance memory interface solution includes the self-calibrating UniPHY megafunction, which is optimized to take advantage of the Stratix V I/O structure and the Quartus® II software TimeQuest Timing Analyzer, which provides the total solution for the highest reliable frequency of operation across process, voltage, and temperature (PVT) variations.

This chapter contains the following sections:

- "Memory Interface Pin Support" on page 7–3
- "Stratix V External Memory Interface Features" on page 7–9

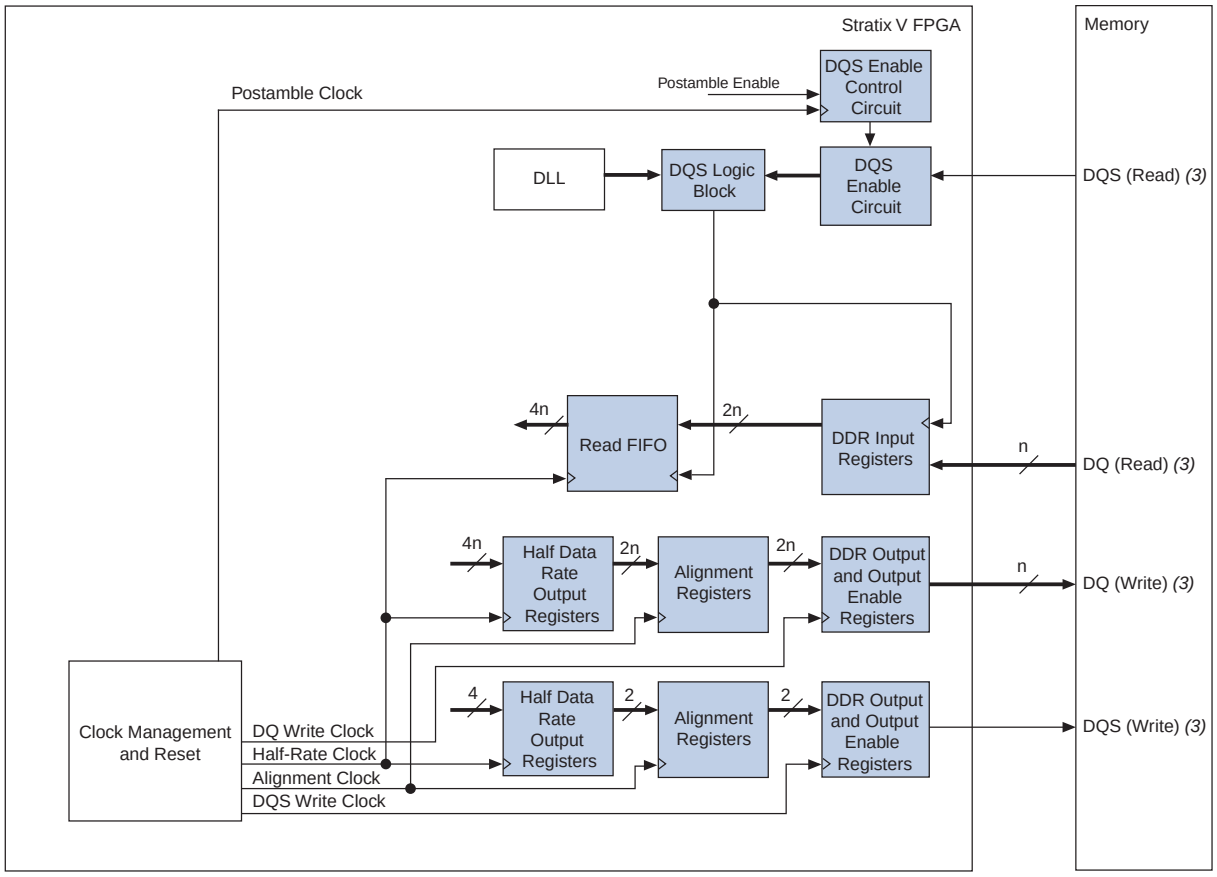
For more information about external memory system performance specifications, board design guidelines, timing analysis, simulation, and debugging information, refer to the *External Memory Interface Handbook*.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Figure 7–1 shows an overview of the memory interface datapath that uses all the Stratix V IOE features.

Figure 7–1. External Memory Interface Datapath Overview for Stratix V Devices (1), (2)



Notes to Figure 7–1:

(1) You can bypass each register block.

(2) The blocks for each memory interface may differ slightly. The shaded blocks are part of the Stratix V IOE.

(3) These signals may be bidirectional or unidirectional, depending on the memory standard. When bidirectional, the signal is active during both read and write operations.

Memory interfaces use Stratix V device features such as delay-locked loops (DLLs), dynamic OCT control, read- and write-leveling circuitry, and I/O features such as programmable I/O delay chains, read FIFO blocks, slew rate adjustment, and programmable drive strength.

For more information about I/O features, refer to the *I/O Features in Stratix V Devices* chapter.

The UniPHY megafunction instantiates a phase-locked loop (PLL) to generate related clocks for the memory interface.

For more information about the Stratix V PLL, refer to the *Clock Networks and PLLs in Stratix V Devices* chapter. For more information about the UniPHY megafunction, refer to *Volume 3: Reference Material* of the *External Memory Interface Handbook*.

Memory Interface Pin Support

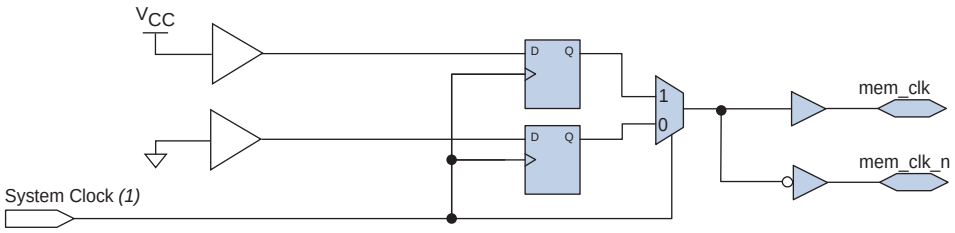
A typical memory interface requires data (D, Q, or DQ), data strobe (DQS/CQ/QK and DQSn/CQn, QK#), address, command, and clock pins. Some memory interfaces use data mask (DM, BWSn, or NWSn) pins to enable write masking and QVLD pins to indicate that the read data is ready to be captured. This section describes how Stratix V devices support all these different pins.

DDR3 and DDR2 SDRAM, and RLDRAM II devices use the CK and CK# signals to capture the address and command signals. Generate these signals to mimic the write-data strobe with Stratix V DDR I/O (DDRIO) registers to ensure that the timing relationships between the CK/CK# and DQS signals (t_{DQSS}, t_{DSS}, and t_{DSH} in DDR3 and DDR2 SDRAM devices or t_{CKDK} in RLDRAM II devices) are met. QDR II+ and QDR II SRAM devices use the same clock (K/K#) to capture write data, address, and command signals.

Memory clock pins in Stratix V devices are generated with a DDRIO register going to differential output pins (refer to Figure 7–2), marked in the pin table with `DIFFOUT`, `DIFFIO_TX`, or `DIFFIO_RX` prefixes.

For more information about which pins to use for memory clock pins and pin location requirements, refer to the *Planning Pin and FPGA Resources* chapter in volume 2 of the *External Memory Interface Handbook*.

Figure 7–2. Memory Clock Generation *(1)*



Note to Figure 7–2:

(1) To minimize jitter, a dedicated clock network is used for memory output clock generation.

Stratix V devices offer differential input buffers for differential read-data strobe and clock operations. In addition, Stratix V devices also provide an independent DQS logic block for each CQn pin for complementary read-data strobe and clock operations. In the Stratix V pin tables, the differential data strobe/clock pin pairs are denoted as DQS and DQSn pins, while the complementary echo clock signals are denoted as CQ and CQn pins. DQSn and CQn pins are marked separately in the pin tables. Each CQn pin connects to a DQS logic block and the phase-shifted CQn signals go to the negative-half cycle input registers in the DQ IOE registers.

Use differential DQS signaling for DDR2 SDRAM interfaces running at or above 333 MHz.

DQ pins can be bidirectional signals, as in DDR3 and DDR2 SDRAM, and RLDRAM II common I/O interfaces, or unidirectional signals, as in QDR II+ and QDR II SRAM, and RLDRAM II separate I/O devices. Connect the unidirectional read-data signals to Stratix V DQ pins and the unidirectional write-data signals to a different DQ/DQS group than the read DQ/DQS group. You must assign the write clocks to the DQS/DQSn pins associated to this write DQ/DQS group. Do not use the CQ/CQn pin-pair for write clocks.

Using a DQ/DQS group for the write-data signals minimizes output skew, allows access to the write-leveling circuitry (for DDR3 SDRAM interfaces), and allows vertical migration. These pins also have access to deskewing circuitry (using programmable delay chains) that can compensate for delay mismatch between signals on the bus.

The DQ and DQS pin locations are fixed in the pin table. Memory interface circuitry is available in every Stratix V I/O bank that does not support transceivers. All the memory interface pins support the I/O standards required to support DDR3 and DDR2 SDRAM, QDR II+ and QDR II SRAM, and RLDRAM II devices.

Stratix V devices support DQ and DQS signals with DQ bus modes of x4, x8/x9, x16/x18, or x32/x36. If any of these pins are not used for memory interfacing, you can use these pins as user I/Os. In addition, you can use the DQSn or CQn pins that are not used for clocking as DQ pins.

Table 7–1 lists pin support per DQ/DQS bus mode, including the DQS/CQ and DQSn/CQn pin pair.

Table 7–1. DQ/DQS Bus Mode Pins for Stratix V Devices

Mode	DQSn Support	CQn Support	Parity or Data Mask (Optional)	QVLD (Optional) (1)	Typical Number of Data Pins per Group	Maximum Number of Data Pins per Group (2)
x4	Yes	No	No (3)	No	4	5
x8/x9 (4)	Yes	Yes	Yes	Yes	8 or 9	11
x16/x18 (5)	Yes	Yes	Yes	Yes	16 or 18	23
x32/x36 (6)	Yes	Yes	Yes	Yes	32 or 36	47

Notes to Table 7–1:

(1) The QVLD pin is not used in the UniPHY megafunction.

(2) This represents the maximum number of DQ pins (including parity, data mask, and QVLD pins) connected to the DQS bus network with single-ended DQS signaling. If you use differential or complementary DQS signaling, the maximum number of data per group decreases by one. This number may vary per DQ/DQS group in a particular device. Check the pin table for the exact number per group. For DDR3 and DDR2 interfaces, the number of pins is further reduced for an interface larger than x8 because you require one DQS pin for each x8/x9 group to form the x16/x18 and x32/x36 groups.

(3) If you do not use differential DQS and the group does not have additional signals, the DM pin is supported.

(4) Two x4 DQ/DQS groups are stitched to create a x8/x9 group, so there are a total of 12 pins in this group.

(5) Four x4 DQ/DQS groups are stitched to create a x16/x18 group; so there are a total of 24 pins in this group.

(6) Eight x4 DQ/DQS groups are stitched to create a x32/x36 group, so there are a total of 48 pins in this group.

Table 7–2 lists the number of DQ/DQS groups available per side in each Stratix V device.

Table 7–2. Number of DQ/DQS Groups in Stratix V Devices per Side [(1)]—Preliminary (Part 1 of 3)

Device	Package	Side	x4 [(2)]	x8/x9	x16/x18	x32/x36
5SGXA3	780-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	26	17	6	1
		Top	34	13	8	2
5SGXA3 5SGXA4	1152-pin FineLine BGA (with 24 transceivers)	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	42	21	10	3
	1152-pin FineLine BGA (with 36 transceivers)	Left/Right	0	0	0	0
		Bottom	36	18	8	2
		Top	36	18	8	2
	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
5SGXA5 5SGXA7	1152-pin FineLine BGA (with 24 transceivers)	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	42	21	10	3
	1152-pin FineLine BGA (with 36 transceivers)	Left/Right	0	0	0	0
		Bottom	36	18	8	2
		Top	36	18	8	2
	1517-pin FineLine BGA (with 36 transceivers)	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
	1517-pin FineLine BGA (with 48 transceivers)	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	50	25	12	5
	1932-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	70	35	16	6
		Top	70	35	16	6
5SGXA9 5SGXAB	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
	1932-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	70	35	16	6
		Top	70	35	16	6

Table 7–2. Number of DQ/DQS Groups in Stratix V Devices per Side (1)—Preliminary (Part 2 of 3)

Device	Package	Side	x4 (2)	x8/x9	x16/x18	x32/x36
5SGXB5 5SGXB6	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	36	18	8	3
		Top	36	18	8	3
	1760-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	50	25	11	4
		Top	50	25	11	4
5SGXB9 5SGXBB	1760-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	50	25	11	3
		Top	50	25	11	3
5SGSD3 5SGSD4	780-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	26	17	6	1
		Top	34	13	8	2
	1152-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	34	17	8	2
		Top	38	19	9	2
5SGSD4	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
5SGSD5	1152-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	42	21	10	3
	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
5SGSD6 5SGSD8	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
	1932-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	70	35	16	6
		Top	70	35	16	6
5SGTC5 5SGTC7	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	50	25	12	5

Table 7–2. Number of DQ/DQS Groups in Stratix V Devices per Side (1)—Preliminary (Part 3 of 3)

Device	Package	Side	x4 (2)	x8/x9	x16/x18	x32/x36
5SEE9 5SEEB	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
	1932-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	70	35	16	6
		Top	70	35	16	6

Notes to Table 7–2:

(1) These numbers are preliminary until the devices are available.

(2) Some of the x4 groups are using RZQ pins. If you use the Stratix V calibrated OCT feature, you cannot use these groups.

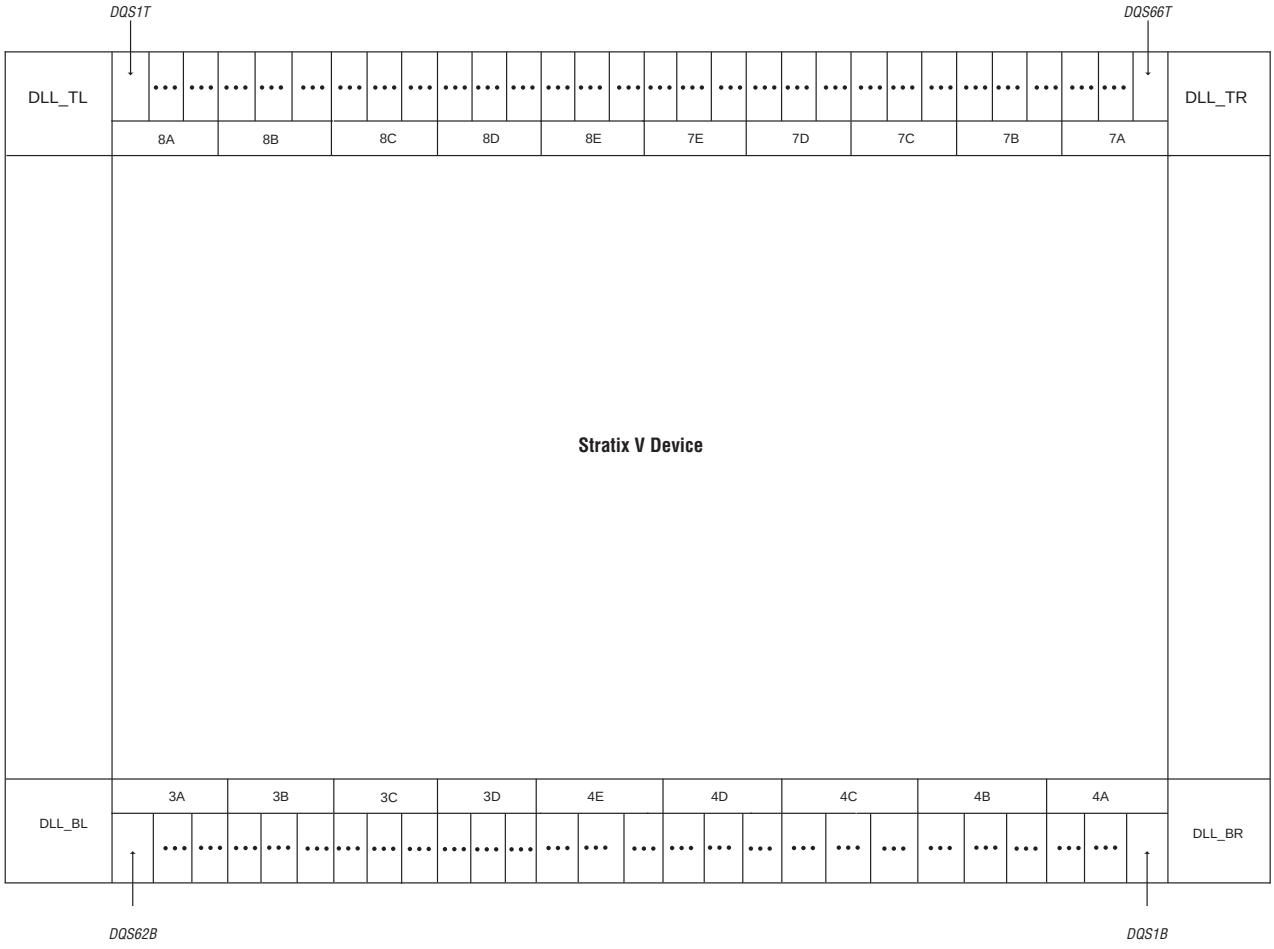
The DQS and DQSn pins are listed in the Stratix V pin tables as DQSXY and DQSnXY, respectively, where X indicates the DQ/DQS grouping number and Y indicates whether the group is located on the top (T), bottom (B), left (L), or right (R) side of the device. The DQ/DQS pin numbering is based on x4 mode.

The corresponding DQ pins are marked as DQXY, where X indicates which DQS group the pins belong to and Y indicates whether the group is located on the top (T) or bottom (B) side of the device. For example, DQS1T indicates a DQS pin located on the top side of the device. The DQ pins belonging to that group are shown as DQ1T in the pin table. For more information, refer to Figure 7–3.

The parity, DM, BWSn, NWSn, ECC, and QVLD pins are shown as DQ pins in the pin table.

The numbering scheme starts from the top-left corner of the device going clockwise in a die-top view. Figure 7–3 shows how the DQ/DQS groups are numbered in a die-top view of the device.

Figure 7–3. DQS Pins in Stratix V I/O Banks



Using the RZQ Pins in a DQ/DQS Group for Memory Interfaces

You can use the DQS/DQSn pins in some of the x4 groups as RZQ pins (listed in the pin table). You cannot use a x4 DQ/DQS group for memory interfaces if any of its DQ pin members are used as RZQ pins for OCT calibration.

There is no restriction on using x8/x9, x16/x18, or x32/x36 DQ/DQS groups that include the x4 groups whose pins are used as RZQ pins because there are enough extra pins that you can use as DQS pins.

☞ You must manually assign DQ and DQS pins for x8, x16/x18, or x32/x36 DQ/DQS groups whose members are used as RZQ pins. The Quartus II software might not be able to place DQ and DQS pins without manual pin assignments, resulting in a "no-fit".

Stratix V External Memory Interface Features

Stratix V devices have features that allow robust high-performance external memory interfacing. The UniPHY megafunction allows you to use these external memory interface features and helps set up the physical interface (PHY) best suited for your system. This section describes each Stratix V device feature that is used in external memory interfaces from the DQS phase-shift circuitry, DQS logic block, leveling multiplexers, and dynamic OCT control block.

☞ The UniPHY megafunction and the Altera memory controller MegaCore® functions can run at half the frequency of the I/O interface of the memory devices to allow better timing management in high-speed memory interfaces. Stratix V devices have built-in circuitries in the IOE to convert data from full rate (the I/O frequency) to half rate (the controller frequency) and vice versa. If you use the Altera memory controller MegaCore functions, the UniPHY megafunction is instantiated for you.

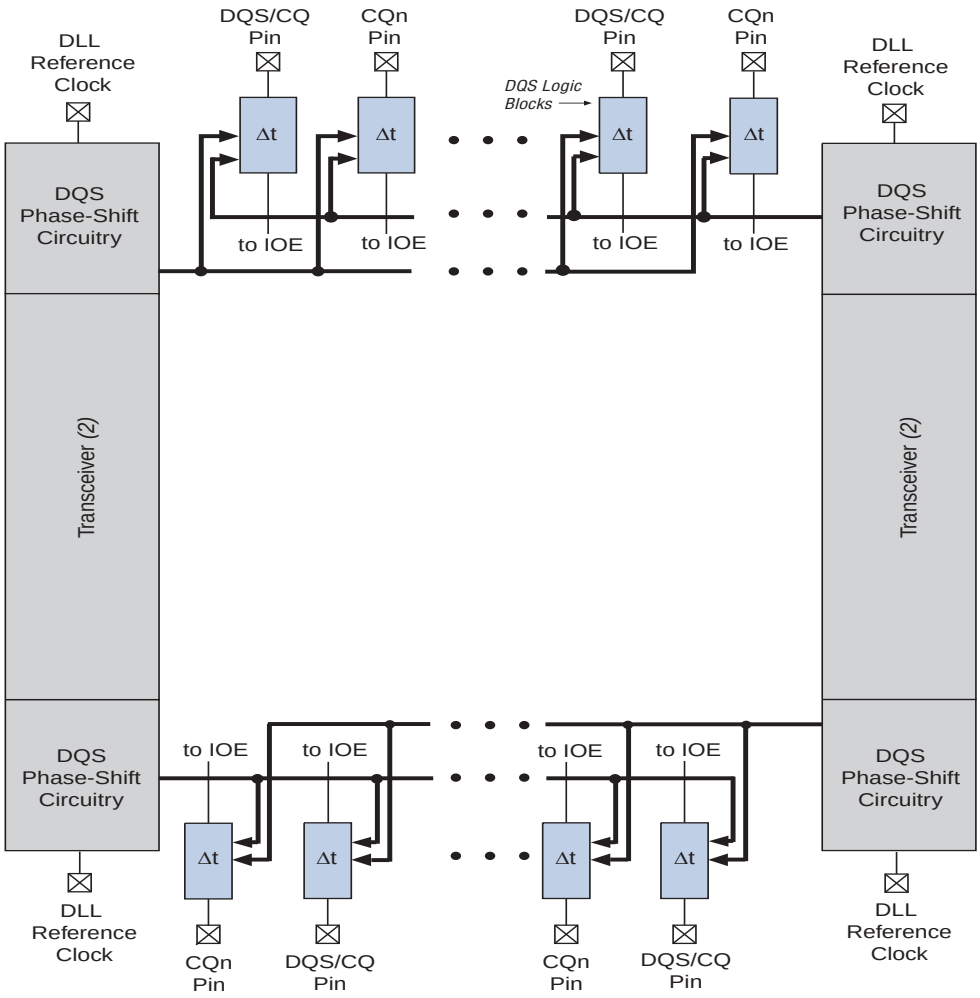
For more information about the UniPHY megafunction, refer to *Volume 3: Reference Material* of the *External Memory Interface Handbook*.

DQS Phase-Shift Circuitry

Stratix V phase-shift circuitry provides phase shift to the DQS/CQ and CQn pins on read transactions if the DQS/CQ and CQn pins are acting as input clocks or strobes to the FPGA. DQS phase-shift circuitry consists of DLLs that are shared between multiple DQS pins and the phase-offset module to further fine-tune the DQS phase shift for different sides of the device.

Figure 7–4 shows how the DQS phase-shift circuitry is connected to the DQS/CQ and CQn pins in the device, where memory interfaces are supported on all sides of the Stratix V device. For possible reference input clock pins for each DLL, refer to "Delay-Locked Loop" on page 7–11.

Figure 7–4. DQS/CQ and CQn Pins and DQS Phase-Shift Circuitry *(1)*



Notes to Figure 7–4:

(1) You can configure each DQS/CQ and CQn pin with a phase shift based on one of two possible DLL output settings.

(2) This transceiver block is applicable to all Stratix V devices except for Stratix V E devices.

DQS phase-shift circuitry is connected to the DQS logic blocks that control each DQS/CQ or CQn pin. DQS logic blocks allow the DQS delay settings to be updated concurrently at every DQS/CQ or CQn pin.

Delay-Locked Loop

DQS phase-shift circuitry uses a DLL to dynamically control the clock delay required by the DQS/CQ and CQn pin. In turn, the DLL uses a frequency reference to dynamically generate control signals for the delay chains in each of the DQS/CQ and CQn pins, allowing it to compensate for PVT variations. The DQS delay settings are Gray-coded to reduce jitter if the DLL updates the settings. If the DLL is in low jitter mode, the phase-shift circuitry requires 2,560 clock cycles to lock and calculate the correct input clock period. Otherwise, only 512 clock cycles are required. Altera recommends not sending data during these clock cycles because there is no guarantee that it will be captured properly. As the settings from the DLL may not be stable until this lock period has elapsed, be aware that anything with these settings (including the leveling delay system) may be unstable during this period.

You can still use DQS phase-shift circuitry for memory interfaces running on frequencies below the minimum DLL input frequency, which is 300 MHz. The frequency of the clock feeding the DLL should be doubled when the interface frequency is between 150 MHz and 299 MHz or multiplied by four when the interface frequency is between 75 MHz and 149 MHz. Because of the changes on the DLL input clock frequency, the DQS delay chain can only shift up to 67.5° for the interface frequency between 150 MHz and 299 MHz and 33.75° for the interface frequency between 75 MHz and 149 MHz. Depending on your design, while the DQS signal might not shift exactly to the middle of the DQ valid window, the IOE is still able to capture the data accurately in low-frequency applications, where a large amount of timing margin is available.

There are a maximum of four DLLs in a Stratix V device, located in each corner of the device. These four DLLs support a maximum of four unique frequencies, with each DLL running at one frequency. Each DLL can have two outputs with different phase offsets, which allows one Stratix V device to have eight different DLL phase shift settings.

Each I/O bank is accessible by two DLLs, giving more flexibility to create multiple frequencies and multiple-type interfaces. Each bank can use settings from either one or both adjacent DLLs. For example, `DQS1T` can get its phase-shift settings from DLL_TR, while `DQS2T` can get its phase-shift settings from DLL_TL.

You can have two different interfaces with the same frequency sharing a DLL, where the DLL controls the DQS delay settings for both interfaces.

If you are using leveling delay chains, you can only have one memory interface in each I/O sub-bank (such as I/O sub-banks 3A, 3B, and 3C). This is because there is only one leveling delay chain per I/O sub-bank.

The reference clock for each DLL may come from the PLL output clocks or clock input pins. Table 7–3 through Table 7–8 list the available DLL reference clock input resources for Stratix V devices.

If you have a dedicated PLL that only generates the DLL input reference clock, set the PLL mode to **No Compensation** to achieve better performance (or the Quartus II software automatically changes it). Because the PLL does not use any other outputs, it does not have to compensate for any clock paths.

Table 7–3. DLL Reference Clock Input for 5SGXA3 (with 36 Transceivers), 5SGXA4, and 5SGSD5 Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X92_Y96 CEN_X92_Y87	COR_X0_Y100 COR_X0_Y91	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X92_Y96 CEN_X92_Y87	COR_X202_Y100 COR_X202_Y91	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P
DLL_BR	CEN_X92_Y11 CEN_X92_Y2	COR_X202_Y10 COR_X202_Y1	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X92_Y11 CEN_X92_Y1	COR_X0_Y10 COR_X0_Y1	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Table 7–4. DLL Reference Clock Input for 5SGXB5 and 5SGXB6 Devices

DLL	PLL		CLKIN		
	Center	Left/Right	Left	Center	Right
DLL_TL	CEN_X90_Y123 CEN_X90_Y114	LR_X0_Y109 LR_X0_Y100	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X90_Y123 CEN_X90_Y114	LR_X197_Y109 LR_X197_Y100	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P
DLL_BR	CEN_X90_Y11 CEN_X90_Y2	LR_X197_Y14 LR_X197_Y5	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X90_Y11 CEN_X90_Y2	LR_X0_Y14 LR_X0_Y5	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Table 7–5. DLL Reference Clock Input for 5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X98_Y118 CEN_X98_Y109	COR_X0_Y122 COR_X0_Y113	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X98_Y118 CEN_X98_Y109	COR_X210_Y122 COR_X210_Y113	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P
DLL_BR	CEN_X98_Y11 CEN_X98_Y2	COR_X210_Y10 COR_X210_Y1	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X98_Y11 CEN_X98_Y2	COR_X0_Y10 COR_X0_Y1	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Table 7–6. DLL Reference Clock Input for 5SGSD3, 5SGSD4, and 5SGXA3 (with 24 Transceivers) Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X84_Y77 CEN_X84_Y68	COR_X0_Y81 COR_X0_Y72	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X84_Y77 CEN_X84_Y68	COR_X185_Y81 COR_X185_Y72	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P
DLL_BR	CEN_X84_Y11 CEN_X84_Y2	COR_X185_Y10 COR_X185_Y1	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X84_Y11 CEN_X84_Y2	COR_X0_Y10 COR_X0_Y1	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Table 7–7. DLL Reference Clock Input for 5SGSD6 and 5SGSD8 Devices

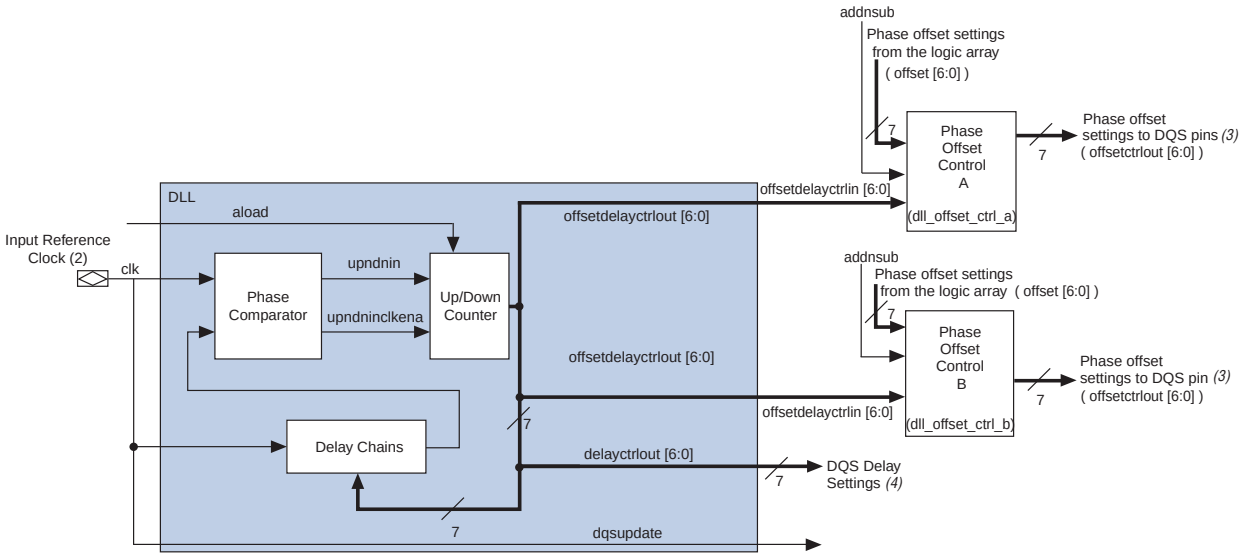
DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X96_Y141 CEN_X96_Y132	COR_X0_Y145 COR_X0_Y136	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X96_Y141 CEN_X96_Y132	COR_X208_Y145 COR_X208_Y136	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P
DLL_BR	CEN_X96_Y11 CEN_X96_Y2	COR_X208_Y10 COR_X208_Y1	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X96_Y11 CEN_X96_Y2	COR_X0_Y10 COR_X0_Y1	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Table 7–8. DLL Reference Clock Input for 5SEE9, 5SEEB, 5SGXA9, 5SGXAB, 5SGXB9, and 5SGXBB Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X104_Y166 CEN_X104_Y157	COR_X0_Y170 COR_X0_Y161	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X104_Y166 CEN_X104_Y157	COR_X225_Y170 COR_X225_Y161	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P
DLL_BR	CEN_X104_Y11 CEN_X104_Y2	COR_X225_Y10 COR_X225_Y1	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X104_Y11 CEN_X104_Y2	COR_X0_Y10 COR_X0_Y1	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Figure 7–5 shows a simple block diagram of the DLL. The input reference clock goes into the DLL to a chain of up to eight delay elements. The phase comparator compares the signal coming out of the end of the delay chain block to the input reference clock. The phase comparator then issues the upndn signal to the Gray-code counter. This signal increments or decrements a seven-bit delay setting (DQS delay settings) that increases or decreases the delay through the delay element chain to bring the input reference clock and the signals coming out of the delay element chain in phase.

Figure 7–5. Simplified Diagram of the DQS Phase-Shift Circuitry *(1)*



Notes to Figure 7–5:

(1) All features of the DQS phase-shift circuitry are accessible from the UniPHY megafunction in the Quartus II software.

(2) The input reference clock for the DQS phase-shift circuitry can come from a PLL output clock or an input clock pin. For more information, refer to Table 7–3 through Table 7–8.

(3) Phase offset settings can only go to the DQS logic blocks.

(4) DQS delay settings can go to the logic array and DQS logic block.

In the Quartus II assignment, the phase offset control block 'A' is designated as DLLOFFSETCTRL_<*coordinate x*>_<*coordinate y*>_N1 and phase offset control block 'B' is designated as DLLOFFSETCTRL_<*coordinate x*>_<*coordinate y*>_N2.

You can reset the DLL from either the logic array or a user I/O pin (when 2,560 or 512 clock cycles applies). Each time the DLL is reset, you must wait for 2,560 (low-jitter mode) or 512 clock cycles for the DLL to lock before you can capture the data properly.

Depending on the DLL frequency mode, the DLL can shift the incoming DQS signals by 0°, 45°, 90°, or 135°. The shifted DQS signal is then used as the clock for the DQ IOE input registers.

All DQS/CQ and CQn pins, referenced to the same DLL, can have their input signal phase shifted by a different degree amount but all must be referenced at one particular frequency. For example, you can have a 90° phase shift on DQS1T and a 45° phase shift on DQS2T, referenced from a 300-MHz clock. However, not all phase-shift combinations are supported. The phase shifts on the DQS pins referenced by the same DLL must all be a multiple of 45° (up to 135°).

The seven-bit DQS delay settings from the DLL vary with PVT to implement the phase-shift delay.

For the frequency range of each mode, refer to the *Stratix V Device Datasheet*.

For a 0° shift, the DQS/CQ signal bypasses both the DLL and DQS logic blocks. The Quartus II software automatically sets the DQ input delay chains, so that the skew between the DQ and DQS/CQ pin at the DQ IOE registers is negligible if a 0° shift is implemented. You can feed the DQS delay settings to the DQS logic block and logic array.

The shifted DQS/CQ signal goes to the DQS bus to clock the IOE input registers of the DQ pins. The signal can also go into the logic array for resynchronization if you are not using IOE resynchronization registers. The shifted CQn signal can only go to the negative-edge input register in the DQ IOE and is only used for QDR II+ and QDR II SRAM interfaces.

Phase Offset Control

Each DLL has two phase-offset modules and can provide two separate DQS delay settings with independent offsets, one for the top and bottom I/O bank and one for the left and right I/O bank, so you can fine-tune the DQS phase-shift settings between two different sides of the device. Even though you have independent phase offset control, the frequency of the interface using the same DLL must be the same. Use the phase offset control module for making small shifts to the input signal and use the DQS phase-shift circuitry for larger signal shifts. For example, if the DLL only offers a multiple of a 45° phase shift, but your interface must have a 97.5° phase shift on the DQS signal, you can use two delay chains in the DQS logic blocks to give you a 90° phase shift and use the phase offset control feature to implement the extra 7.5° phase shift.

You can use either a static phase offset or a dynamic phase offset to implement the additional phase shift. The available additional phase shift is implemented in 2's complement in Gray-code between settings –128 to +127. An additional bit indicates whether the setting has a positive or negative value. The settings are linear, each phase offset setting adds a specified delay amount. The DQS phase shift is the sum of the DLL delay settings and the user-selected phase offset settings whose top setting is 128, so the actual physical offset setting range is 128 subtracted by the DQS delay settings from the DLL.

When using this feature, you must monitor the DQS delay settings to know how many offsets you can add and subtract in the system. The DQS delay settings output by the DLL are also Gray coded.

For example, if the DLL determines that DQS delay settings of 28 is required to achieve a 30° phase shift, you can subtract up to 28 phase offset settings and you can add up to 99 phase offset settings to achieve the optimal delay that you require.

For more information about the value for each step and the specified delay amounts for the phase offset setting, refer to the *Stratix V Device Datasheet*.

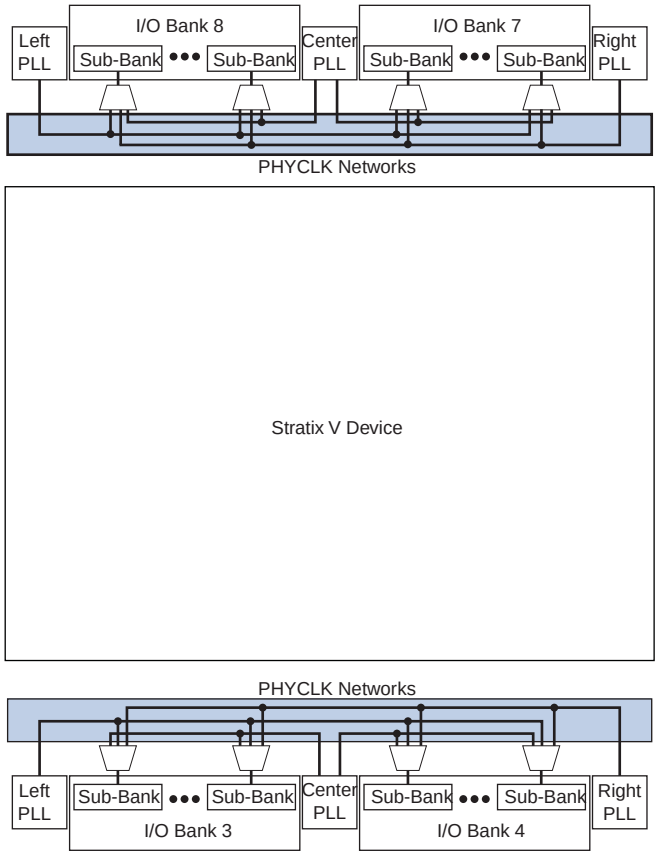
When using static phase offset, you can specify the phase offset amount in the UniPHY megafunction as a positive number for addition or a negative number for subtraction. You can also have a dynamic phase offset that is always added to, subtracted from, or both added to and subtracted from the DLL phase shift. When you add or subtract, you can dynamically input the phase offset amount into the `offset[6..0]` port. When you want to both add and subtract dynamically, you control the `addnsub` signal in addition to the `offset[6..0]` signals. The phase offset is not PVT-compensated.

PHY Clock (PHYCLK) Networks

The PHYCLK network is a dedicated high-speed, low-skew balanced clock tree designed for high-performance external memory interface. The top and bottom side of the Stratix V devices have three PHYCLK networks respectively. Each PHYCLK network spans across one whole side of the device and is driven by either the left, right, or center PLL located at that device side.

The PHYCLK network can be used to drive I/O sub-banks in each I/O bank. Each I/O sub-bank can be driven by only one PHYCLK network. As such, all I/O pins in an I/O sub-bank are driven by the same PHYCLK network. The UniPHY intellectual property (IP) for Stratix V devices uses the PHYCLK network for better performance. Figure 7–6 shows the PHYCLK networks available in Stratix V devices.

Figure 7–6. PHYCLK Networks in Stratix V Devices



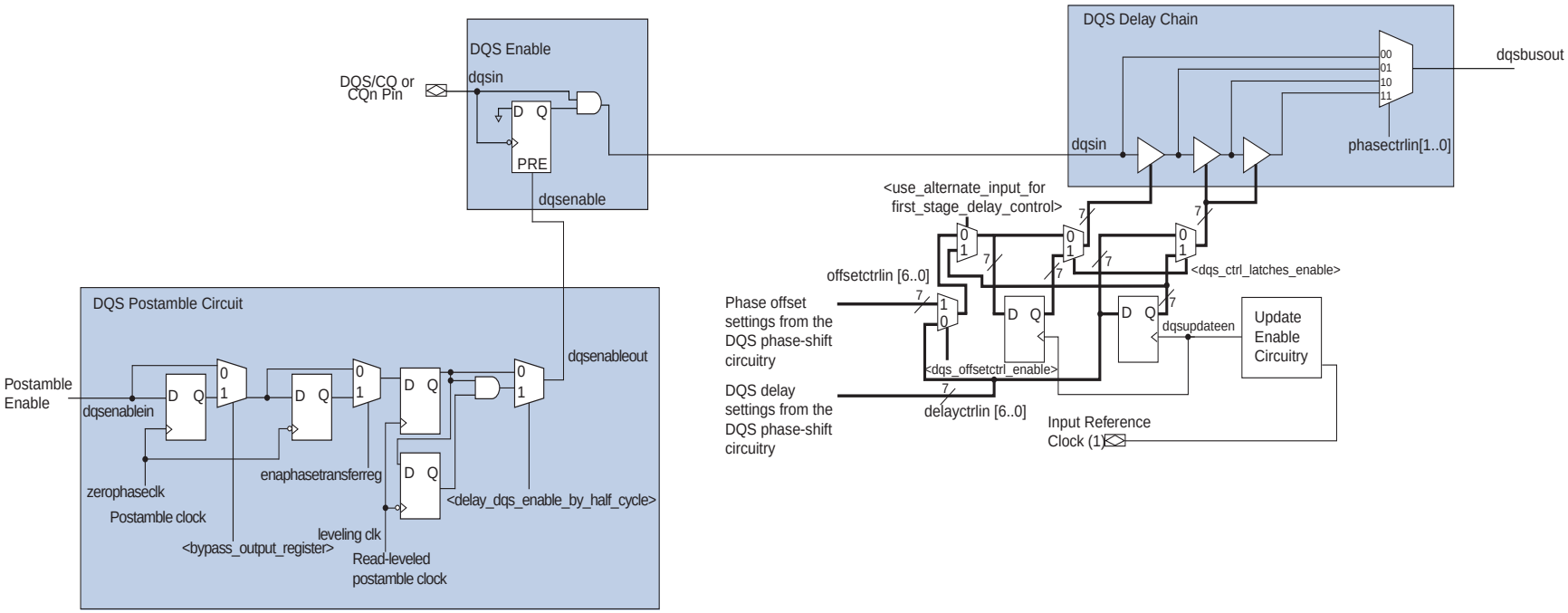
The pin placement guidelines for the PHYCLK networks are as follow:

1. Two interfaces can share an I/O sub-bank (for example, sub-bank 4A) for pin placement if the two interfaces share a PLL. These two interfaces must use the same memory protocol (for example, DDR3), frequency, controller rate (for example, half rate), and phase requirements (for example, additional core-to-periphery clock phase of 90°).
2. Two interfaces that do not share a PLL cannot share a sub-bank for pin placement.
3. Two interfaces can share an I/O bank (for example, I/O bank 4) for pin placement regardless of whether they share a PLL or not.
4. PHYCLK networks do not support split interface, where some pins of a memory interface are placed at the top I/O banks and some pins at the bottom I/O banks.
5. For better performance, you can use the center PLL for your memory interface or you can place all pins for a memory interface in an I/O bank and use the corner PLL adjacent to that I/O bank for the memory interface.
6. The memory interface pins include data, data strobe, data mask, address, command, control, and clock pins.
7. To drive the external memory interfaces, you must use the PLLs in integer PLL mode.

DQS Logic Block

Each DQS/CQ and CQn pin is connected to a separate DQS logic block, which consists of the DQS delay chains, update enable circuitry, and DQS postamble circuitry (Figure 7–7).

Figure 7–7. Stratix V DQS Logic Block



Notes to Figure 7–7:

(1) The input reference clock for the DQS phase-shift circuitry can come from a PLL output clock or an input clock pin. For more information, refer to Table 7–3 on page 7–12 through Table 7–8 on page 7–15.

(2) The `dqsenable` signal can also come from the Stratix V FPGA fabric.

DQS Delay Chain

DQS delay chains consist of a set of variable delay elements to allow the input DQS/CQ and CQn signals to be shifted by the amount specified by the DQS phase-shift circuitry or the logic array. There are four delay elements in the DQS delay chain; the first delay chain closest to the DQS/CQ pin is shifted either by the DQS delay settings or by the sum of the DQS delay setting and the phase-offset setting. The number of delay chains required is transparent because the UniPHY megafunction automatically sets it when you choose the operating frequency. The DQS delay settings can come from the DQS phase-shift circuitry on either end of the I/O banks or from the logic array.

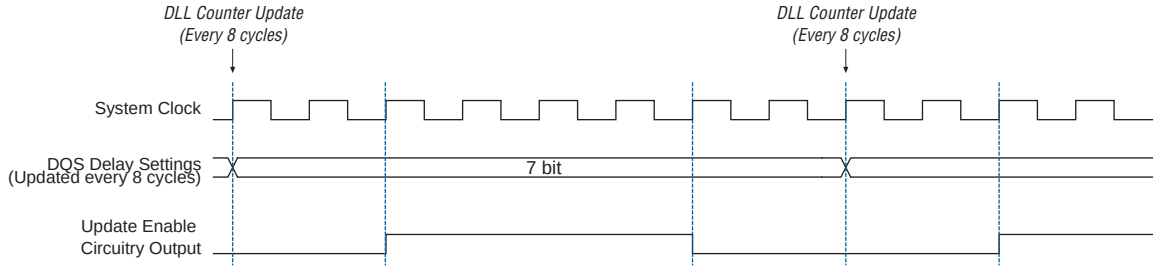
The delay elements in the DQS logic block have the same characteristics as the delay elements in the DLL. When the DLL is not used to control the DQS delay chains, you can input your own Gray-coded 7-bit settings using the `delayctrlin[6..0]` signals available in the UniPHY megafunction. These settings control 1, 2, 3, or all 4 delay elements in the DQS delay chains. The UniPHY megafunction can also dynamically choose the number of DQS delay chains required for the system. The amount of delay is equal to the sum of the intrinsic delay of the delay element and the product of the number of delay steps and the value of the delay steps.

You can also bypass the DQS delay chain to achieve a 0° phase shift.

Update Enable Circuitry

Both the DQS delay settings and the phase-offset settings pass through a register before going into the DQS delay chains. The registers are controlled by the update enable circuitry to allow enough time for any changes in the DQS delay setting bits to arrive at all the delay elements. This allows them to be adjusted at the same time. The update enable circuitry enables the registers to allow enough time for the DQS delay settings to travel from the DQS phase-shift circuitry or core logic to all the DQS logic blocks before the next change. It uses the input reference clock or a user clock from the core to generate the update enable output. The UniPHY megafunction uses this circuit by default. Figure 7–8 shows an example waveform of the update enable circuitry output.

Figure 7–8. DQS Update Enable Waveform



DQS Postamble Circuitry

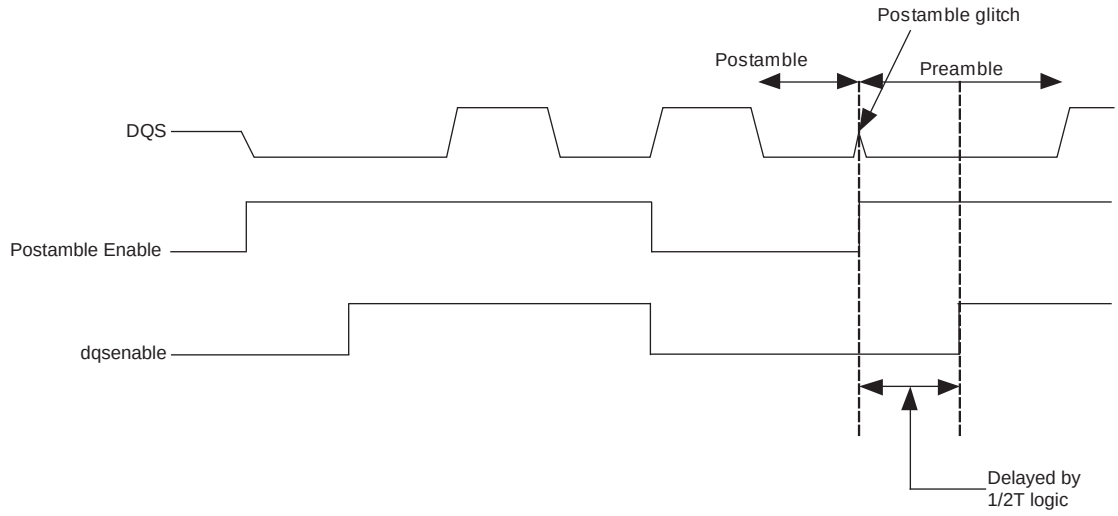
For external memory interfaces that use a bidirectional read strobe (DDR3 and DDR2 SDRAM), the DQS signal is low before going to or coming from a high-impedance state. The state in which DQS is low, just after a high-impedance state, is called the preamble; the state in which DQS is low, just before it returns to a high-impedance state, is called the postamble. There are preamble and postamble specifications for both read and write operations in DDR3 and DDR2 SDRAM. The DQS postamble circuitry ensures that data is not lost if there is noise on the DQS line during the end of a read operation that occurs while DQS is in a postamble state.

Stratix V devices have dedicated postamble registers that you can control to ground the shifted DQS signal used to clock the DQ input registers at the end of a read operation. This ensures that any glitches on the DQS input signals during the end of a read operation that occurs while DQS is in a postamble state do not affect the DQ IOE registers.

In addition to the dedicated postamble register, Stratix V devices also have an HDR block inside the postamble enable circuitry. Use these registers if the controller is running at half the frequency of the I/Os.

Using the HDR block as the first stage capture register in the postamble enable circuitry block is optional. The HDR block is clocked by the half-rate resynchronization clock, which is the output of the I/O clock divider circuit. There is an AND gate after the postamble register outputs to avoid postamble glitches from a previous read burst on a non-consecutive read burst. This scheme allows half-a-clock cycle latency for `dqsenable` assertion and zero latency for `dqsenable` deassertion (refer to Figure 7–9).

Figure 7–9. Avoiding Glitch on a Non-Consecutive Read Burst Waveform

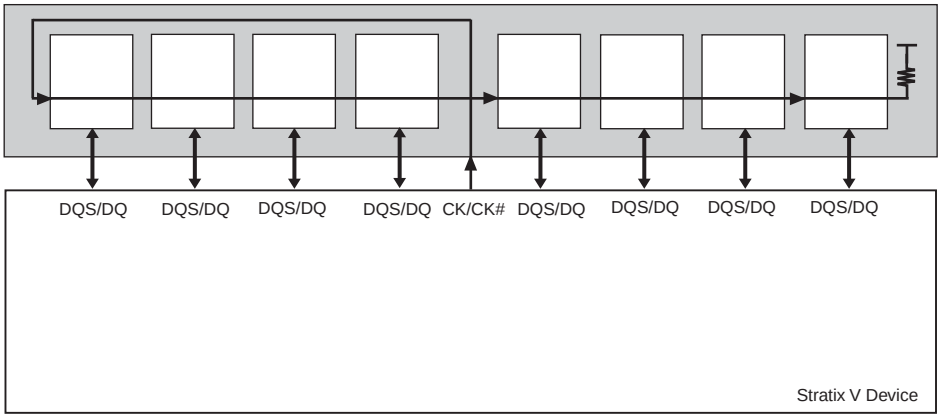


Leveling Circuitry

DDR3 SDRAM unbuffered modules use a fly-by clock distribution topology for better signal integrity. This means that the CK/CK# signals arrive at each DDR3 SDRAM device in the module at different times. The difference in arrival time between the first DDR3 SDRAM device and the last device on the module can be as long as 1.6 ns.

Figure 7–10 shows the clock topology in DDR3 SDRAM unbuffered modules.

Figure 7–10. DDR3 SDRAM Unbuffered Module Clock Topology



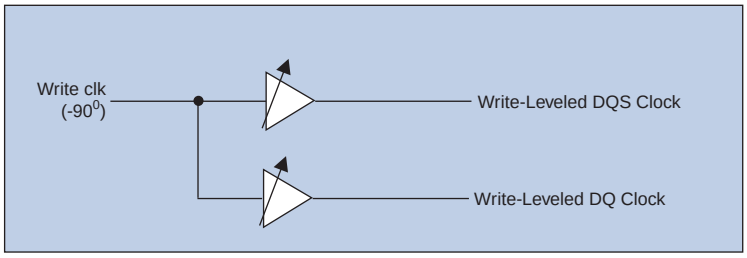
Because the data and read strobe signals are still point-to-point, take special care to ensure that the timing relationship between the CK/CK# and DQS signals (tDQSS, tDSS, and tDSH) during a write is met at every device on the modules. In a similar way, read data coming back into the FPGA from the memory is also staggered.

Stratix V FPGAs have leveling circuitry to address these two situations. There is one leveling circuit per I/O sub-bank (for example, I/O sub-bank 1A, 1B, and 1C each has one leveling circuitry). These delay chains are PVT-compensated by the same DQS delay settings as the DLL and DQS delay chains.

The DLL uses eight delay chains, such that each delay chain generates a 45° delay. The generated clock phases are distributed to every DQS logic block that is available in the I/O sub-bank. The delay chain taps then feed a multiplexer controlled by the UniPHY megafunction to select which clock phases are to be used for that x4 or x 8 DQS group. Each group can use a different tap output from the read-leveling and write-leveling delay chains to compensate for the different CK/CK# delay going into each device on the module.

Figure 7–11 shows the Stratix V write-leveling circuitry.

Figure 7–11. Write-Leveling Delay Chains and Multiplexers for Stratix V Devices (1)



Note to Figure 7–11:

(1) There is one leveling delay chain per I/O sub-bank (for example, I/O sub-banks 1A, 1B, and 1C). You can only have one memory interface in each I/O sub-bank when you use the leveling delay chain.

The –90° write clock of the UniPHY megafunction feeds the write-leveling circuitry to produce the clock to generate the DQS and DQ signals. During initialization, the UniPHY megafunction picks the correct write-leveled clock for the DQS and DQ clocks for each DQ/DQS group after sweeping all the available clocks in the write calibration process. The DQ clock output is –90° phase-shifted compared to the DQS clock output.

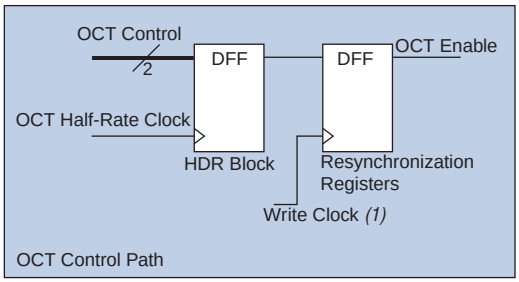
The UniPHY megafunction dynamically calibrates the alignment for read and write leveling during the initialization process.

For more information about the UniPHY megafunction, refer to *Volume 3: Reference Material* of the *External Memory Interface Handbook*.

Dynamic On-Chip Termination Control

Figure 7–12 shows the dynamic OCT control block.

Figure 7–12. Dynamic OCT Control Block for Stratix V Devices



Note to Figure 7–12:

(1) The write clock comes from either the PLL or the write-leveling delay chain.

The block includes all the registers required to dynamically turn on-chip parallel termination (R_T OCT) on during a read and turn R_T OCT off during a write.

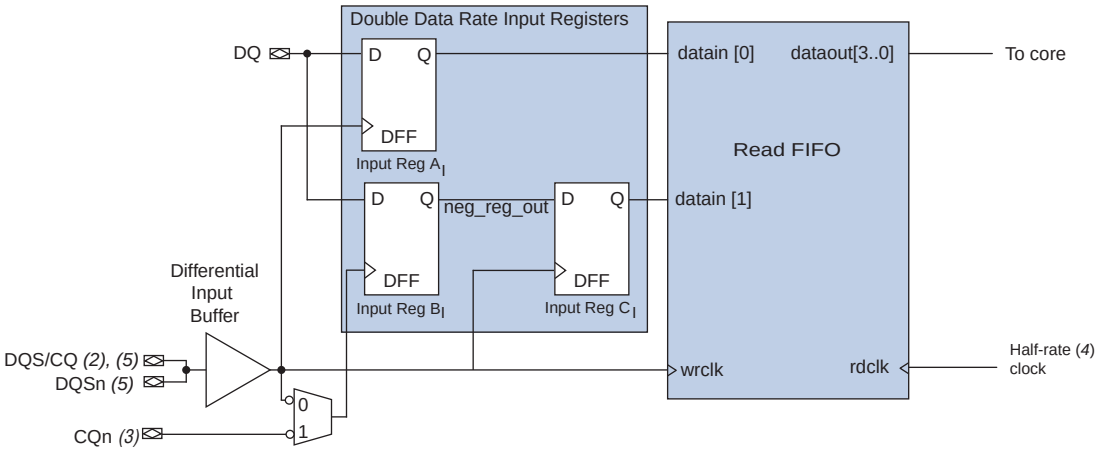
For more information about dynamic OCT control, refer to the *I/O Features in Stratix V Devices* chapter.

I/O Element Registers

The IOE registers are expanded to allow source-synchronous systems to have faster register-to-register transfers and resynchronization. Both top, bottom, and right IOEs have the same capability.

Figure 7–13 shows the registers available in the Stratix V input path. The input path consists of the DDR input registers and the read FIFO block. You can bypass each block of the input path.

Figure 7–13. IOE Input Registers for Stratix V Devices *(1)*



Notes to Figure 7–13:

(1) You can bypass the register or read FIFO block in this path.
(2) The input clock can be from the DQS logic block or from a global clock line.
(3) This input clock comes from the CQn logic block.
(4) This half rate-read clock comes from a PLL through the clock network.
(5) The DQS and DQSn signals must be inverted for DDR3 and DDR2 SDRAM interfaces. When using Altera's memory interface IPs, the DQS and DQSn signals are automatically inverted.

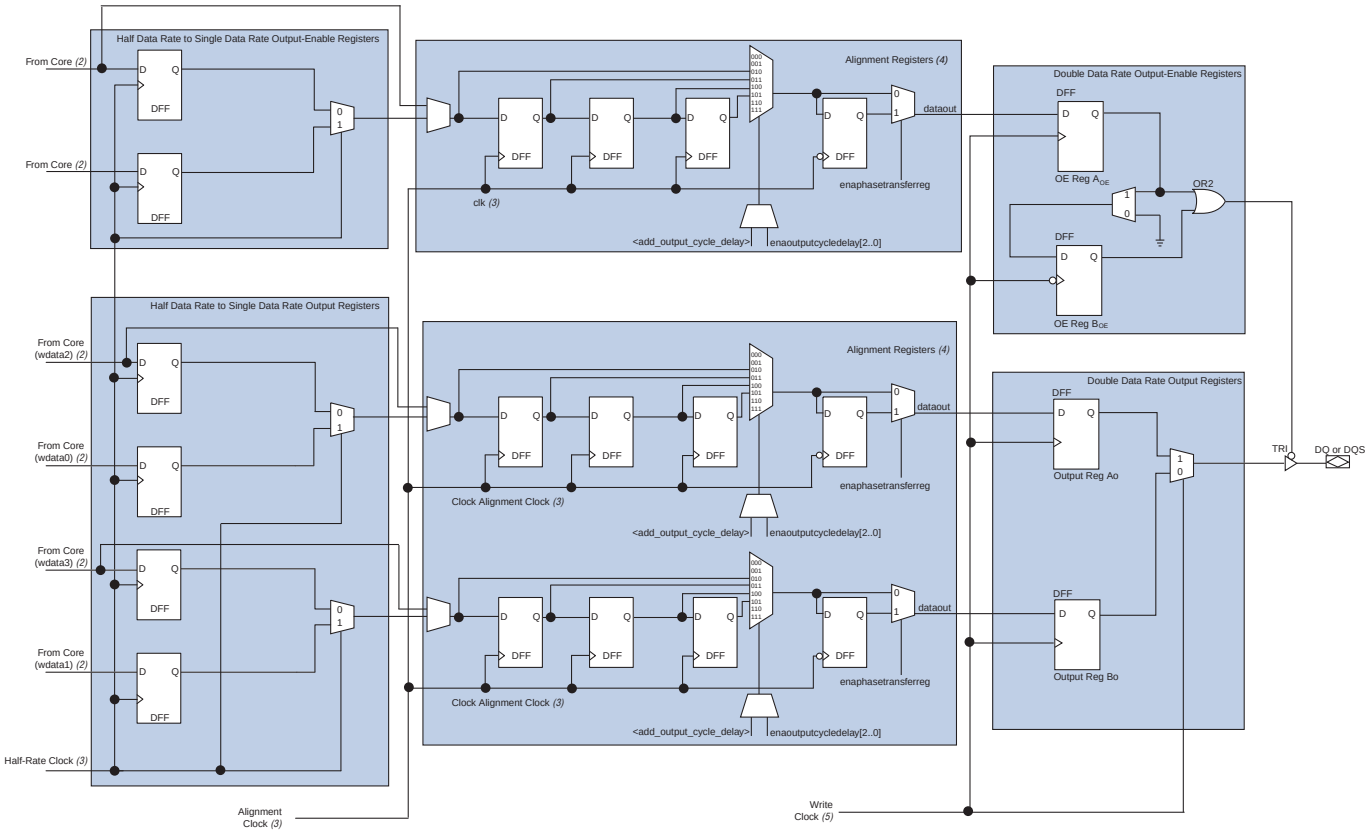
There are three registers in the DDR input registers block. Two registers capture data on the positive and negative edges of the clock while the third register aligns the captured data. You can choose to use the same clock for the positive and negative edge registers or two complementary clocks (DQS/CQ for the positive-edge register and DQSn/CQn for the negative-edge register). The third register that aligns the captured data uses the same clock as the positive edge registers.

The read FIFO block resynchronizes the data to the system clock domain, as well as to lower the data rate to half rate.

For more information about the read-leveling delay chain, refer to "Leveling Circuitry" on page 7–23.

Figure 7–14 shows the registers available in the Stratix V output and output-enable paths. The path is divided into the HDR block, alignment registers, and output and output-enable registers. The device can bypass each block of the output and output-enable path.

Figure 7–14. IOE Output and Output-Enable Path Registers for Stratix V Devices *(1)*



Notes to Figure 7–14:

(1) You can bypass each register block of the output and output-enable paths.

(2) Data coming from the FPGA core are at half the frequency of the memory interface clock frequency in half-rate mode.

(3) The half-rate clock comes from the PLL while the alignment clock comes from the write-leveling delay chains.

(4) These registers are used in DDR3 SDRAM interfaces for write-leveling purposes.

(5) The write clock can come from either the PLL or from the write-leveling delay chain. The DQ write clock and DQS write clock have a 90° offset between them.

The output path is designed to route combinatorial or registered single data rate (SDR) outputs and full-rate or half-rate DDR outputs from the FPGA core. Half-rate data is converted to full-rate with the HDR block, clocked by the half-rate clock from the PLL. The resynchronization registers are also clocked by the same 0° system clock, except in the DDR3 SDRAM interface. In DDR3 SDRAM interfaces, the leveling registers are clocked by the write-leveling clock.

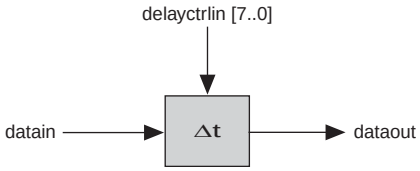
For more information about the write-leveling delay chain, refer to "Leveling Circuitry" on page 7–23.

The output-enable path has a structure similar to the output path. You can have a combinatorial or registered output in SDR applications and you can use half-rate or full-rate operation in DDR applications. Also, the output-enable path's resynchronization registers have a structure similar to the output path registers, ensuring that the output-enable path goes through the same delay and latency as the output path.

Delay Chain

Stratix V devices have run-time adjustable delay chains in the I/O blocks and the DQS logic blocks. You can control the delay chain setting through the I/O or the DQS configuration block output. Figure 7–15 shows the delay chain ports.

Figure 7–15. Delay Chain

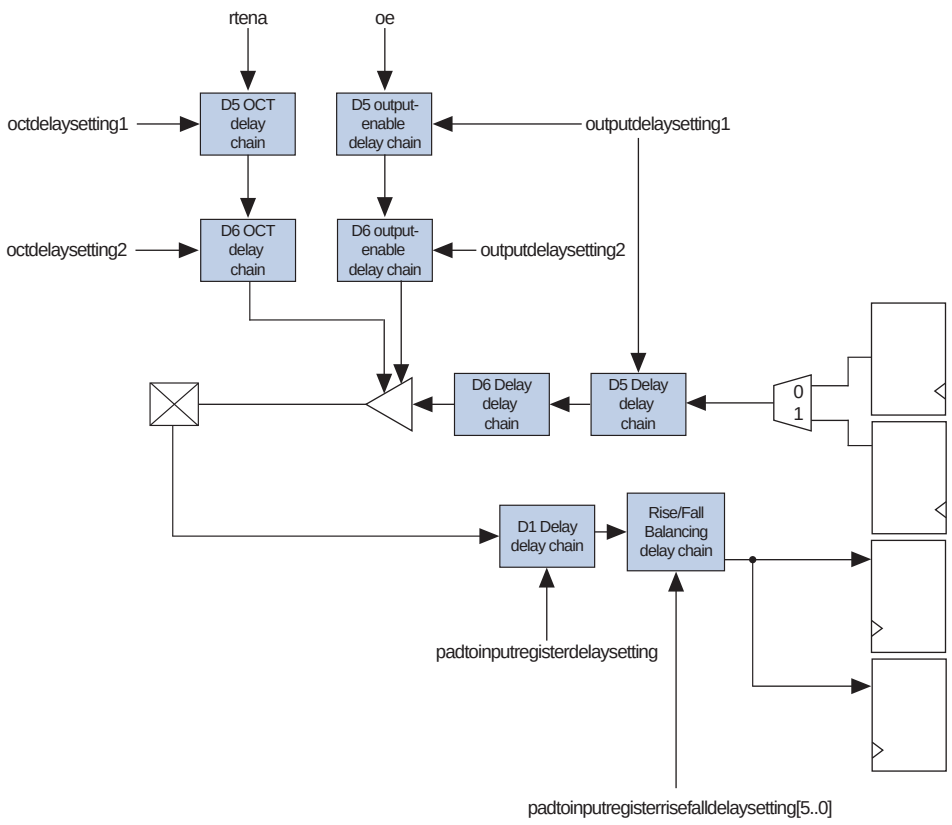


Every I/O block contains the following:

- Two delay chains in series between the output registers and output buffer
- Two delay chains between the input buffer and input register
- Two delay chains between the output enable and output buffer
- Two delay chains between the R_T OCT enable-control register and output buffer

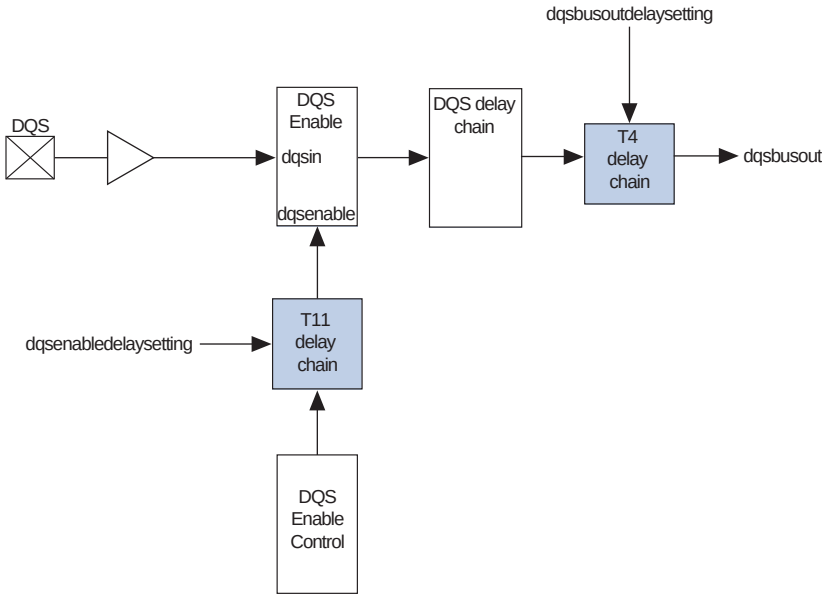
Figure 7–16 shows the delay chains in an I/O block.

Figure 7–16. Delay Chains in an I/O Block



Each DQS logic block contains a delay chain after the dqsbusout output and another delay chain before the dqsenable input. Figure 7–17 shows the delay chains in the DQS input path.

Figure 7–17. Delay Chains in the DQS Input Path



I/O Configuration Block and DQS Configuration Block

The I/O configuration block and the DQS configuration block are shift registers that you can use to dynamically change the settings of various device configuration bits. The shift registers power-up low. Every I/O pin contains one I/O configuration register while every DQS pin contains one DQS configuration block in addition to the I/O configuration register. Figure 7–18 shows the I/O configuration block and the DQS configuration block circuitry.

Figure 7–18. I/O Configuration Block and DQS Configuration Block

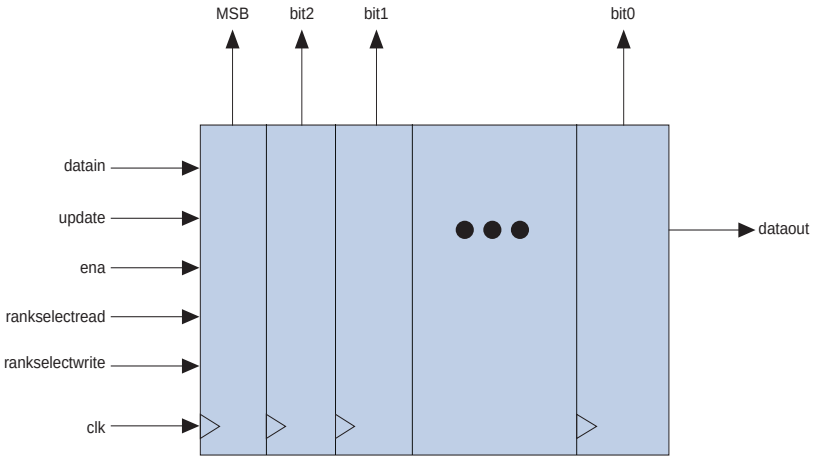


Table 7–9 lists the I/O configuration block bit sequence.

Table 7–9. I/O Configuration Block Bit Sequence

Bit	Bit Name
0..5	padtoinputregisterdelaysetting[0..5]
6..11	padtoinputregisterrisefalldelaysetting[0..5]
12..17	outputdelaysetting1[0..5]
18..23	outputdelaysetting2[0..5]
24..25	inputclkndelaysetting[0..1]
26..27	inputclkdelaysetting[0..1]
28..32	Not mapped to any port
33	dutycycledelaymode
34	Not mapped to any port
35..38	dutycycledelaysetting[0..3]
39	Not mapped to any port

Table 7–10 lists the DQS configuration block bit sequence.

Table 7–10. DQS Configuration Block Bit Sequence (Part 1 of 2)

Bit	Bit Name
0..5	dqsbusoutdelaysetting[0..5]
6..11	dqsbusoutdelaysetting2[0..5]
12..17	octdelaysetting1[0..5]
18..23	octdelaysetting2[0..5]
24..25	addrphasesetting[0..1]
26	addrpowerdown
27	addrphaseinvert
28..29	dqsoutputphasesetting[0..1]
30	dqsoutputpowerdown
31	dqsoutputphaseinvert
32..33	dqoutputphasesetting[0..1]
34	—
35	dqoutputpowerdown
36	dqoutputphaseinvert
37..38	resyncinputphasesetting[0..1]
39	resyncinputpowerdown
40	resyncinputphaseinvert
41..42	postamblephasesetting[0..1]
43	—
44	postamblepowerdown
45	postamblephaseinvert
46..47	dqs2xoutputphasesetting[0..1]

Table 7–10. DQS Configuration Block Bit Sequence (Part 2 of 2)

Bit	Bit Name
48	dqs2xoutputpowerdown
49	dqs2xoutputphaseinvert
50..51	dq2xoutputphasesetting[0..1]
52	—
53	dq2xoutputpowerdown
54	dq2xoutputphaseinvert
55..56	ck2xoutputphasesetting[0..1]
57	ck2xoutputpowerdown
58	ck2xoutputphaseinvert
59..60	dqoutputzerophasesetting[0..1]
61..62	postamblezerophasesetting[0..1]
63	postamblepowerdown
64	dividerioehratephaseinvert
65	dividerphaseinvert
66..68	enaoctcycledelaysetting[0..2]
69	enaoctphasetransferreg
70..77	dqsdisablendelaysetting[0..7]
78..85	dqsenabledelaysetting[0..7]
86	enadqsenablephasetransferreg
87..88	dqsinputphasesetting[0..1]
89	enadqsphasetransferreg
90	enaoutputphasetransferreg
91..93	enadqscycledelaysetting[0..2]
94..96	enaoutputcycledelaysetting[0..2]
97	enainputcycledelaysetting
98	enainputphasetransferreg
99	—
100	—

Document Revision History

Table 7–11 lists the revision history for this chapter.

Table 7–11. Document Revision History

Date	Version	Changes
June 2012	1.4	■ Added Table 7–6, Table 7–8, and Table 7–9. ■ Updated Table 7–2, Table 7–3, and Table 7–7. ■ Updated Figure 7–18. ■ Updated the "PHY Clock (PHYCLK) Networks" section.
November 2011	1.3	■ Added "PHY Clock (PHYCLK) Networks" section. ■ Updated "Delay-Locked Loop" section. ■ Updated Figure 7–3, Figure 7–5, and Figure 7–7. ■ Updated Table 7–2, Table 7–3, Table 7–4, Table 7–5, and Table 7–6. ■ Minor text edits.
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Figure 7–4, Figure 7–6, Figure 7–13, Figure 7–14, and Figure 7–17. ■ Updated Table 7–2, Table 7–7, and Table 7–8. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

Section III. System Integration

This section provides information about system integration in Stratix® V devices. This section includes the following chapters:

- Chapter 8, Hot Socketing and Power-On Reset in Stratix V Devices
- Chapter 9, Configuration, Design Security, and Remote System Upgrades in Stratix V Devices
- Chapter 10, SEU Mitigation in Stratix V Devices
- Chapter 11, JTAG Boundary-Scan Testing in Stratix V Devices
- Chapter 12, Power Management in Stratix V Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

8. Hot Socketing and Power-On Reset in Stratix V Devices

SV51009-1.4

This chapter provides information about hot-socketing specifications, power sequencing, power-on reset (POR) requirements, and their implementation in Stratix® V devices.

Stratix V devices offer hot socketing, also known as hot plug-in or hot swap. You can insert or remove a Stratix V device or a board in a system during system operation without causing undesirable effects to the running system bus or board that is inserted into the system, provided that the device's power sequencing requirement is met.

The hot-socketing feature also removes some of the difficulty when using Stratix V devices on PCBs that contain a mixture of 3.0-, 2.5-, 1.8-, 1.5-, 1.35-, 1.25-, 1.20-, and 0.85-V devices.

The Stratix V hot-socketing feature provides the following:

- Board or device insertion and removal without external components or board manipulation
- I/O buffers non-intrusive to system buses during hot insertion

This chapter contains the following sections:

- "Stratix V Hot-Socketing Specifications" on page 8–1
- "Hot-Socketing Feature Implementation in Stratix V Devices" on page 8–3
- "Power Sequencing Requirements" on page 8–4
- "Power-On Reset Circuitry" on page 8–5
- "Power-On Reset Specifications" on page 8–7

Stratix V Hot-Socketing Specifications

Stratix V devices are hot-socketing compliant without the need for external components or special design requirements. The advantages of hot-socketing support in Stratix V devices include the following:

- "Stratix V Devices Can Be Driven Before Power Up"
- "I/O Pins Remain Tri-Stated During Power Up"
- "Insertion or Removal of a Stratix V Device from a Powered-Up System"

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Stratix V Devices Can Be Driven Before Power Up

You can drive signals into I/O pins, dedicated input pins, and dedicated clock pins of Stratix V devices before or during power up or power down without damaging the device. External input signals to I/O pins of the device do not internally power the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies of the device through internal paths within the device.

I/O Pins Remain Tri-Stated During Power Up

A device that does not support hot socketing can interrupt the system operation or cause contention by driving out before or during power up. In a hot-socketing situation, the Stratix V output buffers are turned off during system power up or power down. Also, the Stratix V device does not drive out until the device is configured and working within the recommended operating conditions.

Insertion or Removal of a Stratix V Device from a Powered-Up System

When powered up through the device signal pins, devices that do not support hot socketing are directly connected to the GND, causing power supplies failure. This irregular power up can damage both the driving and driven devices and can disrupt card power up.

You can insert a Stratix V device into or remove it from a powered-up system board without damaging the system board or interfering with its operation.

The hotsocket circuit monitors the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies. During hot socketing, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20 pF.

You must ensure that all power supplies for Stratix V devices are within the ramp-up and ramp-down rate of 200 µs to 100 ms.

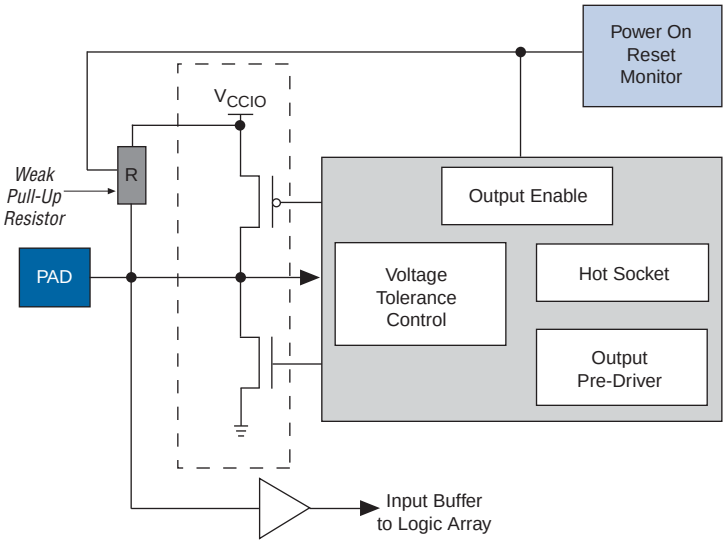
A possible concern regarding hot socketing is the potential for “latch up”. When hot socketing, Stratix V devices are immune to latch up. Latch up occurs when electrical subsystems are hot socketed into an active system. During hot socketing, the signal pins can be connected and driven by the active system before the power supply can provide current to the power and GND planes of the device. This condition can lead to latch up and cause a low-impedance path from power to GND within the device. As a result, the device draws a large amount of current, possibly causing electrical damage.

Hot-Socketing Feature Implementation in Stratix V Devices

The hot-socketing feature turns off the output buffer during power up and power down of the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies. When the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies are below the threshold voltage, the hot-socketing circuitry generates an internal HOTSCKT signal. Hot-socketing circuitry is designed to prevent excess I/O leakage during power up. When the voltage ramps up very slowly, it is still relatively low, even after the POR signal is released and the configuration is completed. The CONF_DONE and nSTATUS pins fail to respond, as the output buffer cannot flip from the state set by the hot-socketing circuit at this low voltage. Therefore, the hot-socketing circuitry is removed from these configuration pins to ensure that they can operate during configuration. Thus, it is an expected behavior for these pins to drive out during power-up and power-down sequences.

Figure 8–1 shows the I/O pin circuitry for Stratix V devices.

Figure 8–1. Hot-Socketing Circuitry for Stratix V Devices



The POR circuit monitors the voltage level of the power supplies (V_{CC}, V_{CCPGM}, V_{CCPD}, V_{CC_AUX}, V_{CCBAT}, and V_{CCPT}) and keeps the I/O pins tri-stated until the device is in user mode. The weak pull-up resistor (R) in the Stratix V input/output element (IOE) keeps the I/O pins from floating. The 3.0-V tolerance control circuit permits 3.0 V to drive the I/O pins before the V_{CC}, V_{CCPT}, V_{CCPD}, and V_{CCIO} power supplies are powered and prevents the I/O pins from driving out when the device is not in user mode.

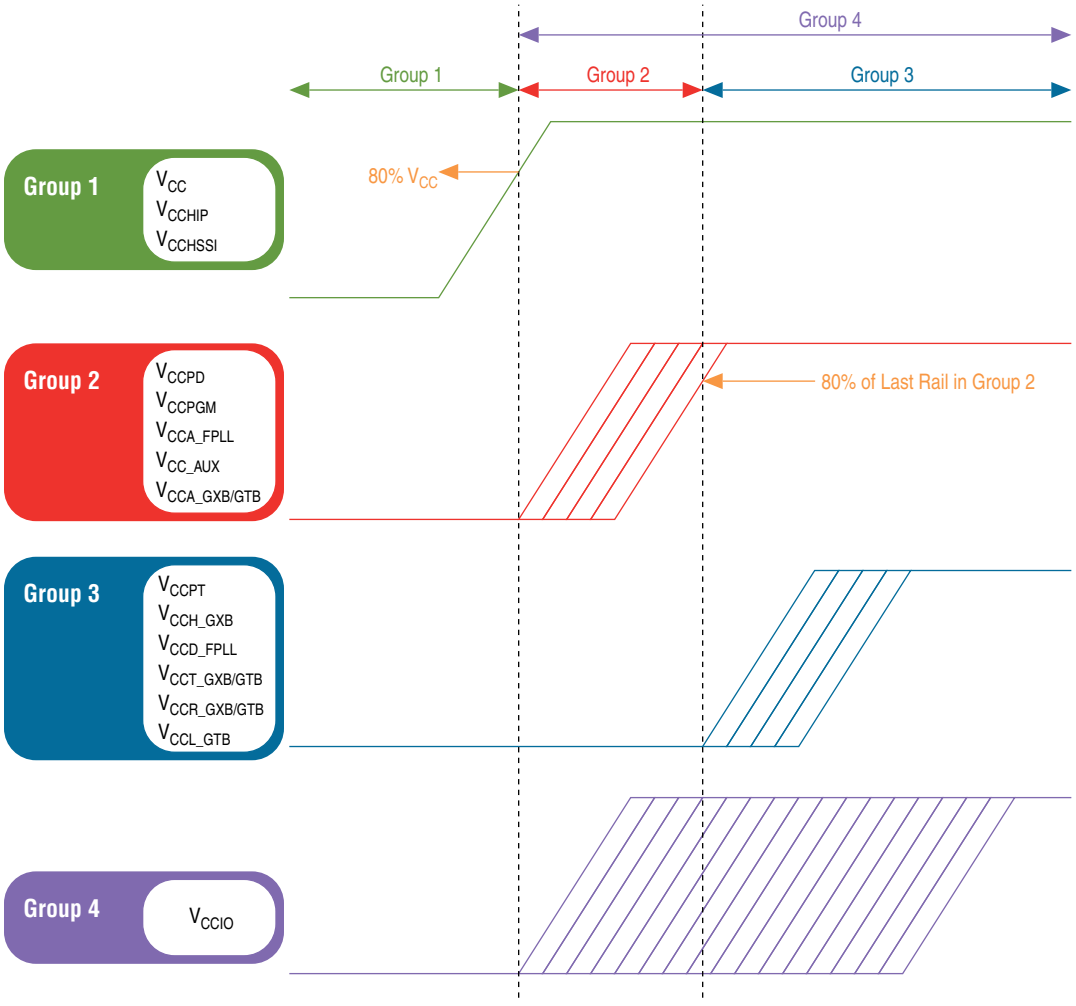
Altera uses GND as a reference for hot-socketing operations and I/O buffer designs. To ensure proper operation, you must connect the GND between boards before connecting the power supplies. This prevents the GND on your board from being pulled up inadvertently by a path to power through other components on your board. A pulled up GND could otherwise cause an out-of-specification I/O voltage or over current condition with the Altera® device.

For the V_{CC_AUX} power supply, POR only monitors one of the VCC_AUX pins. You must connect all the VCC_AUX pins.

Power Sequencing Requirements

Stratix V devices require a power-up sequence as shown in Figure 8–2 to prevent excessive inrush current. This power-up sequence is divided into four power groups. Group 1 contains the first power rails to ramp. The V_{CC}, V_{CCHIP}, and V_{CCHSSI} power rails in this group must ramp to a minimum of 80% of their full rail before any other power rails may start. Group 1 power rails can continue to ramp to full rail. The power rails in Group 2 and Group 4 can start to ramp in any order after Group 1 has reached its minimum 80% threshold. When the last power rail in Group 2 reaches 80% of its full rail, the remaining power rails in Group 3 may start their ramp. During this time, Group 2 power rails may continue to ramp to full rail. Power rails in Group 3 may ramp in any order. All power rails must ramp monotonically. The complete power-up sequence must meet either the standard or fast POR delay time, depending on the POR delay setting that is used.

Figure 8–2. Power-Up Sequence Requirement for Stratix V Devices *(1), (2), (3)*



Notes to Figure 8–2:

(1) V_{CCBAT} can be powered up at any time.
(2) If V_{CC}, V_{CCR_GXB}, and V_{CCT_GXB} have the same voltage level, they can be powered by the same regulator in Group 1 and ramp simultaneously.
(3) All rails can start power-down simultaneously but must reach 0 V within 100 ms.

Stratix V devices may power down all power rails simultaneously. However, all rails must reach 0 V within 100 ms from the start of power-down.

Power-On Reset Circuitry

This section describes POR circuitry in Stratix V devices. POR circuitry keeps the devices in the reset state until the power supply outputs are within operating range.

When power is applied to a Stratix V device, a POR event occurs if the power supply reaches the recommended operating range within the maximum power supply ramp time (t_{RAMP}). If t_{RAMP} is not met, the device I/O pins and programming registers remain tri-stated, during which device configuration could fail. The maximum t_{RAMP} for Stratix V devices is 100 ms; the minimum t_{RAMP} is 200 µs. All supplies must be strictly monotonic—without plateaus. Set the POR delay by selecting the `MSEL` pins. The standard or fast POR delay mode that is generated internally from the `MSEL` decoding block determines the POR delay time, as follows:

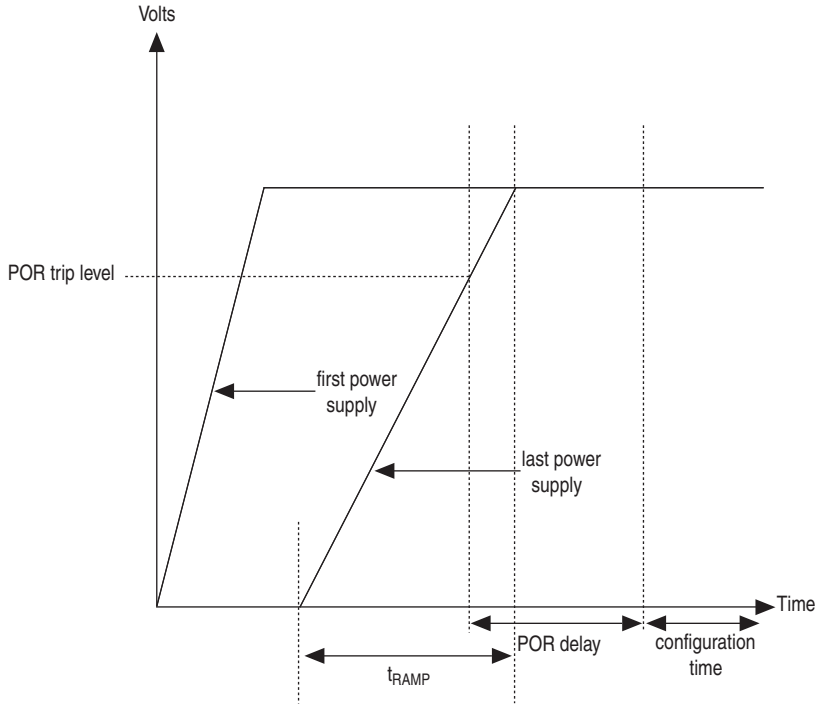
- For the standard POR delay mode, the POR delay time is 100 to 300 ms.
- For the fast POR delay mode, the POR delay time is 4 to 12 ms.
- With fast POR, the maximum POR delay width is 12 ms, which leaves enough time after the POR trip for the PCI Express® (PCIe®) to initialize.

To use Configuration via Protocol (CvP), the total time must be within 10 ms, from the first power supply ramp-up to the last power supply ramp-up.

For more information about the `MSEL` pin settings, refer to the *Configuration, Design Security, and Remote System Upgrades in Stratix V Devices* chapter.

Figure 8–3 shows the relationship between t_{RAMP} and POR delay.

Figure 8–3. Relationship between t_{RAMP} and POR Delay



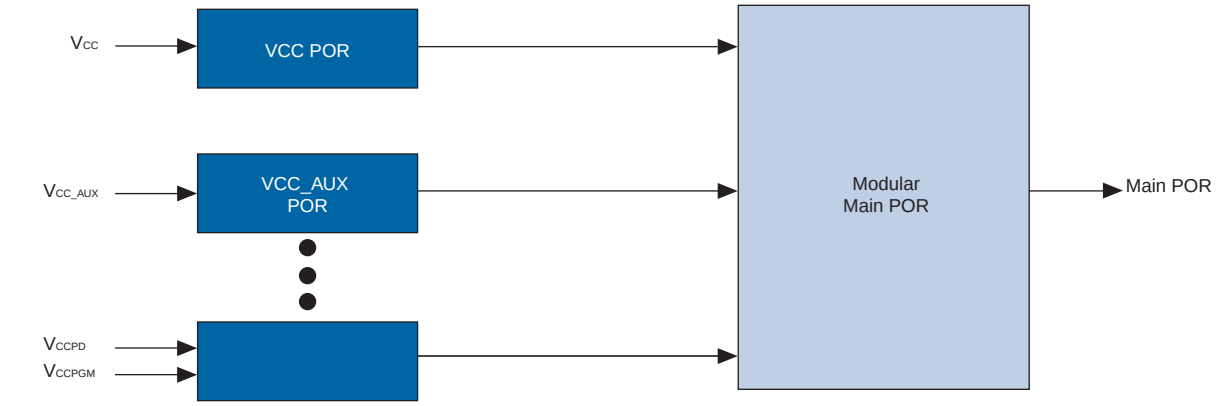
The Stratix V POR implements a modularized design structure to monitor the supplies gating the main POR. Each external power supply being monitored has a base POR, which employs a generic design to detect the corresponding voltage level for proper functionality.

It also checks for functionality of I/O level shifters powered by the V_{CCPD} and V_{CCPGM} power supplies during power-up mode. The main POR waits for all the individual PORs to release the POR signal so that the control block can start programming the device. The Stratix V main POR also employs a modularized structure to enable brown-out detection on a power supply and also to possibly expand the number of power supplies being monitored.

All configuration-related dedicated and dual function I/O pins must be powered by the V_{CCPGM} power supply.

Figure 8–4 shows a simplified POR diagram for Stratix V devices.

Figure 8–4. Simplified POR Diagram for Stratix V Devices



Power-On Reset Specifications

Table 8–1 lists the power supplies that the POR monitors.

Table 8–1. Power Supplies Monitored by the POR Circuitry

Power Supply	Description	Setting (V)
V_{CC_AUX}	Auxiliary supply for the programmable power technology	2.5
V_{CCBAT}	Battery back-up power supply for design security volatile key register	3.0–1.2
V_{CC}	Core and periphery power supply	0.85
V_{CCPT}	Programmable power technology power supply	1.5
V_{CCPD}	I/O pre-driver power supply	2.5, 3.0
V_{CCPGM}	Configuration pins power supply	1.8, 2.5, 3.0

Table 8–2 lists the power supplies that the POR does not monitor.

Table 8–2. Power Supplies Not Monitored by the POR Circuitry

Power Supply	Description	Setting (V)
V_{CCT_GXB}	Transmitter power	0.85, 1.0
V_{CCH_GXB}	Transmitter output buffer power	1.5
V_{CCR_GXB}	Receiver power	0.85, 1.0
V_{CCA_GXB}	Transceiver high voltage power	2.5, 3.0
V_{CCHIP}	Transceiver HIP digital power	0.85
V_{CCIO}	I/O power supply	1.2, 1.25, 1.35, 1.5, 1.8, 2.5, 3.0
V_{CCA_FPLL}	PLL analog global power supply	2.5
V_{CCD_FPLL}	PLL digital power supply	1.5

For more information about the POR specification, refer to the *Stratix V Device Datasheet*.

Document Revision History

Table 8–3 lists the revision history for this chapter.

Table 8–3. Document Revision History

Date	Version	Changes
June 2012	1.4	Minor text edits.
November 2011	1.3	■ Added “Power Sequencing Requirements” section. ■ Updated Figure 8–3 and Figure 8–4. ■ Updated Table 8–1. ■ Updated “Hot-Socketing Feature Implementation in Stratix V Devices” and “Power-On Reset Circuitry” sections.
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated “Power-On Reset Circuitry” section. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

9. Configuration, Design Security, and Remote System Upgrades in Stratix V Devices

SV51010-1.7

This chapter contains information about the Stratix® V supported configuration schemes, instructions about how to execute the required configuration schemes, and all the necessary option pin settings. This chapter also reviews the different ways you can configure your device and explains the design security and remote system upgrade features for the Stratix V devices.

This chapter includes the following sections:

- "Configuration Features" on page 9–2
- "Power-On Reset Circuit and Configuration Pins Power Supply" on page 9–2
- "Configuration Sequence" on page 9–4
- "Configuration Schemes" on page 9–7
- "Fast Passive Parallel Configuration" on page 9–9
- "Active Serial Configuration (Serial Configuration Devices)" on page 9–17
- "Passive Serial Configuration" on page 9–29
- "JTAG Configuration" on page 9–36
- "Device Configuration Pins" on page 9–40
- "Configuration Data Decompression" on page 9–44
- "Remote System Upgrades" on page 9–46
- "Design Security" on page 9–55

Stratix V devices use SRAM cells to store configuration data. Because SRAM memory is volatile, you must download the configuration data to the Stratix V device each time the device powers up. You can configure Stratix V devices using one of four configuration schemes:

- Fast passive parallel (FPP) (×8, ×16, and ×32)
- Active serial (AS) (×1 and ×4)
- Passive serial (PS)
- JTAG

All configuration schemes use either an external controller (for example, a MAX® II device, MAX V device, or microprocessor), a configuration device, or a download cable. For more information about the configuration features, refer to "Configuration Features".

For more information about the Configuration via Protocol (CvP) scheme, refer to the *Configuration via Protocol (CvP) Implementation in Altera FPGAs User Guide*.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Configuration Features

Stratix V devices offer decompression, design security, and remote system upgrade features. Stratix V devices can receive a compressed configuration bitstream and decompress this data in real-time, reducing storage requirements and configuration time. Design security using configuration bitstream encryption is available in Stratix V devices, which protects your designs. You can make real-time system upgrades of your Stratix V designs from remote locations with the remote system upgrade feature.

Table 9–1 lists which configuration features you can use in each configuration scheme.

Table 9–1. Configuration Features for Stratix V Devices

Configuration Scheme	Decompression	Design Security	Remote System Upgrade
FPP (×8, ×16, ×32)	Y (1)	Y (1)	—
AS (×1, ×4)	Y	Y	Y
PS	Y	Y	—
JTAG	—	—	—

Note to Table 9–1:

(1) In these configuration schemes, the host system must accommodate a different DCLK-to-DATA[] ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

Power-On Reset Circuit and Configuration Pins Power Supply

The following sections describe the power-on reset (POR) circuit and the power supply for the configuration pins.

POR Delay Specification

POR delay is defined as the delay between the time when all the power supplies monitored by the POR circuitry reach the minimum recommended operating voltage to the time when the nSTATUS is released high and your device is ready to begin configuration.

For more information about the POR delay, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

Table 9–2 lists the fast and standard POR delay specification.

Table 9–2. Fast and Standard POR Delay Specification (1)

POR Delay	Minimum	Maximum
Fast	4 ms	12 ms
Standard	100 ms	300 ms

Note to Table 9–2:

(1) You can select the POR delay based on the MSEL settings as described in Table 9–4 on page 9–7.

Power-On Reset Circuit

The POR circuit keeps the entire system in reset mode until the power supply voltage levels have stabilized on power-up. After power-up, the device does not release nSTATUS until all the power supplies monitored by the POR circuitry are above the device's POR trip point. On power down, brown-out occurs if any of the power supplies monitored by the POR circuitry drops below the threshold level of the hot-socket circuitry.

For more information about which power supplies are monitored by the POR circuitry, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

V_{CCPGM} Pin

Stratix V devices have a power supply, V_{CCPGM}, for all dedicated configuration pins and dual-purpose pins. The supported configuration voltages are 1.8, 2.5, and 3.0 V.

Use the V_{CCPGM} pin to power all dedicated configuration inputs, dedicated configuration outputs, dedicated configuration bidirectional pins, and the dual-purpose pins that you use for configuration. The configuration input buffers do not have to share power lines with the regular I/O buffer in Stratix V devices.

The operating voltage for the configuration input pin is independent of the I/O banks power supply, V_{CCIO}, during configuration. Therefore, Stratix V devices do not require configuration voltage constraints on V_{CCIO}.

For more information about the configuration pins connections, refer to the *Stratix V Device Family Pin Connection Guidelines*.

V_{CCPD} Pin

Stratix V devices have a dedicated programming power supply, V_{CCPD}, which must be connected to 3.0 V or 2.5 V to power the I/O pre-drivers and JTAG I/O pins (TCK, TMS, TDI, TDO, and TRST).

V_{CCPD} must be greater than or equal to V_{CCIO}. If V_{CCIO} is set to 3.0 V, V_{CCPD} must be powered up to 3.0 V. If the V_{CCIO} of the bank is set to 2.5 V or lower, V_{CCPD} must be powered up to 2.5 V. This applies for all the banks containing the VCCPD and VCCIO pins.

For more information about the configuration pins power supply, refer to "Device Configuration Pins" on page 9–40.

Configuration Sequence

The following sections describe the general configuration process for the FPP, AS, and PS schemes.

Power Up

To begin the configuration process, you must fully power-up all the power supplies monitored by the POR circuitry to the appropriate voltage levels. The power supplies must ramp-up monotonically within the specified ramp-up time to ensure successful configuration.

All power supplies including the V_{CCPGM} and V_{CCPD} must ramp-up from 0 V to the desired voltage level within the ramp-up time specification. If these supplies are not ramped up within this specified time, your Stratix V device will not configure successfully. If your system cannot ramp-up the power supplies within the specified ramp-up time specification, you must hold nCONFIG low until all the power supplies are stable.

For more information about the ramp-up time specification, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

Reset

After power-up, the Stratix V device goes through a POR. The POR delay depends on the MSEL settings. During POR, the device resets, holds nSTATUS low, clears the configuration RAM bits, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins remain tri-stated until the device is configured.

While nCONFIG is low, the device is in reset. When the device comes out of reset, nCONFIG must be at a logic-high level in order for the device to release the open-drain nSTATUS pin. After nSTATUS is released, it is pulled high by a pull-up resistor and the device is ready to receive configuration data. Before and during configuration, all user I/O pins are tri-stated. If nIO_pullup is driven low during power up and configuration, the user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are on after the device exits POR, before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

For more information about the POR delay specification, refer to "POR Delay Specification" on page 9–2.

Configuration

Both nCONFIG and nSTATUS must be deasserted at a logic-high level in order for the configuration stage to begin. For the FPP and PS configuration schemes, the device receives configuration data on its DATA pins and the clock source on the DCLK pin. Configuration data is latched into the Stratix V device on the rising edge of DCLK. For the AS configuration scheme, the device receives configuration data on its AS_DATA[] pins and drives the clock source on the DCLK pin. Configuration data is latched into the Stratix V device on the falling edge of DCLK.

After the Stratix V device has received all the configuration data successfully, it releases the CONF_DONE pin, which is pulled high by a pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration has completed and initialization of the device can begin. For the FPP and PS schemes, DCLK must not be left floating at the end of configuration. You must drive them either high or low, whichever is convenient on your board.

For the FPP and PS schemes, there is no maximum DCLK period, which means you can stop the configuration by holding the DCLK low for an indefinite amount of time. To resume configuration, the external host must provide data on the DATA[] pins prior to sending the first DCLK rising edge.

Configuration Error

If the **Auto-restart configuration after error** option (available in the Quartus® II software from the **General** panel of the **Device and Pin Options** dialog box) is turned on, the Stratix V device releases the nSTATUS pin high after the specified time indicated by tSTATUS and retries the configuration. If this option is turned off or if you are using a PS or FPP scheme with an external controller, the system must monitor the nSTATUS for errors and sends a low-to-high signal on nCONFIG for specified t_{CFG} time to restart the configuration.

Initialization

In Stratix V devices, initialization begins after the CONF_DONE goes high. For the FPP and PS configuration schemes, two DCLK falling edges are required after the last configuration byte is sent to the Stratix V device to begin the initialization of the device for both uncompressed and compressed configuration data.

The initialization clock source is from the internal oscillator, CLKUSR, or DCLK pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix V device provides itself with enough clock cycles for proper initialization.

Table 9–3 lists the initialization clock source option, the applicable configuration schemes, and the maximum frequency.

Table 9–3. Initialization Clock Source Option and the Maximum Frequency

Initialization Clock Source	Configuration Schemes	Maximum Frequency	Minimum Number of Clock Cycles (1)
Internal Oscillator	AS, PS, FPP	12.5 MHz	17,408 (3)
CLKUSR	AS, PS, FPP (2)	125 MHz	

Notes to Table 9–3:

(1) The minimum number of clock cycles required for device initialization.

(2) To enable CLKUSR as the initialization clock source, turn on the **Enable user-supplied start-up clock (CLKUSR)** option in the Quartus II software from the **General** panel of the **Device and Pin Options** dialog box.

(3) The number is still preliminary.

If you use the optional CLKUSR pin as the initialization clock source and nCONFIG is pulled low to restart configuration during device initialization, ensure that CLKUSR or DCLK continues toggling until nSTATUS goes low and goes high again.

CLKUSR provides you with the flexibility to synchronize initialization of multiple devices or to delay initialization. Supplying a clock on the CLKUSR pin during initialization does not affect configuration. After CONF_DONE goes high, CLKUSR or DCLK is enabled after the time specified by t_{CD2CU}. When this time period elapses, Stratix V devices require a minimum number of clock cycles to initialize properly and enter user mode as specified by the t_{CD2UMC} parameter.

User Mode

The Stratix V device enters user mode when the initialization is complete. You can monitor the end of the initialization stage by enabling the optional INIT_DONE pin. If enabled, the low-to-high transition of INIT_DONE indicates the device has completed initialization and entered user mode. In this mode, your design is executed. The user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

At any time during the configuration stage or user mode operation, you can initiate a reconfiguration by setting a low pulse on the nCONFIG pin. The pulse must meet the minimum t_{CFG} low-pulse width. When nCONFIG is pulled low, the nSTATUS and CONF_DONE pins are pulled low and all I/O pins are tri-stated. Configuration begins when the nCONFIG and nSTATUS pins return to a logic-high level.

Configuration Schemes

The following sections describe configuration schemes for Stratix V devices.

MSEL Pin Settings

Select the configuration scheme by driving the Stratix V device MSEL pins either high or low (refer to Table 9–4). The MSEL input buffers are powered by the V_{CCPGM} power supply. During POR and during reconfiguration, the MSEL pins must be at the LVTTL V_{IL} and V_{IH} levels to be considered logic low and logic high, respectively.

To avoid problems with detecting an incorrect configuration scheme, hardwire the MSEL pins to V_{CCPGM} or GND without pull-up or pull-down resistors. Use only the MSEL pin settings listed in Table 9–4. Do not drive the MSEL pins with a microprocessor or another device.

Table 9–4 lists the configuration schemes for Stratix V devices.

Table 9–4. Configuration Schemes for Stratix V Devices (1) (Part 1 of 2)

Configuration Scheme	Decompression Feature	Design Security Feature	Configuration Voltage Standard (V) (3)	POR Delay (6)	MSEL[4..0]
FPP ×8	Disabled	Disabled	1.8/2.5/3.0	Fast	10100
				Standard	11000
	Disabled	Enabled	1.8/2.5/3.0	Fast	10101
				Standard	11001
	Enabled	Optional (2)	1.8/2.5/3.0	Fast	10110
				Standard	11010
FPP ×16	Disabled	Disabled	1.8/2.5/3.0	Fast	00000
				Standard	00100
	Disabled	Enabled	1.8/2.5/3.0	Fast	00001
				Standard	00101
	Enabled	Optional (2)	1.8/2.5/3.0	Fast	00010
				Standard	00110
FPP ×32	Disabled	Disabled	1.8/2.5/3.0	Fast	01000
				Standard	01100
	Disabled	Enabled	1.8/2.5/3.0	Fast	01001
				Standard	01101
	Enabled	Optional (2)	1.8/2.5/3.0	Fast	01010
				Standard	01110
PS	Optional (2)	Optional (2)	1.8/2.5/3.0	Fast	10000
				Standard	10001
AS (×1, ×4) (4)	Optional (2)	Optional (2)	3.0	Fast	10010
				Standard	10011

Table 9–4. Configuration Schemes for Stratix V Devices (1) (Part 2 of 2)

Configuration Scheme	Decompression Feature	Design Security Feature	Configuration Voltage Standard (V) (3)	POR Delay (6)	MSEL[4..0]
JTAG-based configuration (5)	Disabled	Disabled	—	—	(7)

Notes to Table 9–4:

(1) Do not leave the MSEL pins floating. Connect them to V_{CCPGM} or GND. Use only the MSEL pin settings listed in Table 9–4.

(2) You can select to enable or disable this feature.

(3) The configuration voltage standard applied to the V_{CCPGM} power supply that powers all the configuration pins during configuration.

(4) The AS configuration scheme supports the remote system upgrade feature. For more information about the remote system upgrade feature, refer to "Remote System Upgrades" on page 9–46.

(5) JTAG-based configuration takes precedence over other configuration schemes. This means the MSEL pin settings are ignored. JTAG-based configuration does not support the design security or decompression features.

(6) For POR delay specification, refer to "POR Delay Specification" on page 9–2.

(7) Altera recommends connecting the MSEL pins to GND.

Raw Binary File Size

For the POR delay specification, refer to "POR Delay Specification" on page 9–2.

Table 9–5 lists the uncompressed raw binary file (**.rbf**) sizes for Stratix V devices.

Table 9–5. Uncompressed .rbf Sizes for Stratix V Devices (1)

Family	Device	Configuration .rbf Size (bits)	IOCSR .rbf Size (bits)
Stratix V GX	5SGXA3	213,798,720	1,689,600
	5SGXA4	213,798,720	1,689,600
	5SGXA5	269,978,848	2,239,436
	5SGXA7	269,978,848	2,239,436
	5SGXA9	387,394,048	2,141,440
	5SGXAB	387,394,048	2,141,440
	5SGXB5	270,528,480	2,236,416
	5SGXB6	270,528,480	2,236,416
	5SGXB9	387,394,048	2,141,440
	5SGXBB	387,394,048	2,141,440
Stratix V GT	5SGTC5	269,978,848	2,195,200
	5SGTC7	269,978,848	2,195,200
Stratix V GS	5SGSD3	93,080,448	933,120
	5SGSD4	209,935,224	2,004,480
	5SGSD5	209,935,224	2,004,480
	5SGSD6	266,798,896	2,236,416
	5SGSD8	266,798,896	2,236,416
Stratix V E (2)	5SEE9	387,394,048	2,141,440
	5SEEB	387,394,048	2,141,440

Notes to Table 9–5:

(1) These values are preliminary.

(2) Stratix V E devices do not have PCI Express® (PCIe®) hard IP. Stratix V E devices do not support the CvP configuration scheme.

Use the data in Table 9–5 to estimate the file size before design compilation. Different configuration file formats, such as a hexadecimal (**.hex**) or tabular text file (**.ttf**) format, have different file sizes. For the different types of configuration file and file sizes, refer to the Quartus II software. However, for a specific version of the Quartus II software, any design targeted for the same device has the same uncompressed configuration file size. If you are using compression, the file size can vary after each compilation because the compression ratio depends on your design.

For more information about setting device configuration options or creating configuration files, refer to the *Device Configuration Options* and *Configuration File Formats* chapters in volume 2 of the *Configuration Handbook*.

Fast Passive Parallel Configuration

The FPP configuration using an external host provides the fastest method to configure Stratix V devices. FPP is supported in multiple data widths—8-bits, 16-bits, and 32-bits. You can perform a FPP configuration of Stratix V devices using an external host such as a MAX II device, MAX V device, or microprocessor. The external host controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix V device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** formats. Therefore, the design that controls the configuration stages, such as fetching the data from flash memory and sending it to the device, must be stored in the MAX II device, MAX V device, or microprocessor.

The Parallel Flash Loader (PFL) feature in MAX II and MAX V devices provides an efficient method to program CFI flash memory devices through the JTAG interface. PFL also acts as a controller to read configuration data from the flash memory device and configures the Stratix V device. PFL supports both the PS and FPP configuration schemes.

For more information about the PFL, refer to *Parallel Flash Loader Megafunction User Guide*.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device for both uncompressed and compressed configuration data in a FPP configuration.

DCLK-to-DATA[] Ratio for FPP configuration

FPP configuration requires a different DCLK-to-DATA[]ratio when you enable the design security, decompression, or both features. Table 9–6 lists the DCLK-to-DATA[]ratio for each combination.

Table 9–6. DCLK-to-DATA[] Ratio (1) (Part 1 of 2)

Configuration Scheme	Decompression	Design Security	DCLK-to-DATA[] Ratio
FPP ×8	Disabled	Disabled	1
	Disabled	Enabled	1
	Enabled	Disabled	2
	Enabled	Enabled	2

Table 9–6. DCLK-to-DATA[] Ratio (1) (Part 2 of 2)

Configuration Scheme	Decompression	Design Security	DCLK-to-DATA[] Ratio
FPP ×16	Disabled	Disabled	1
	Disabled	Enabled	2
	Enabled	Disabled	4
	Enabled	Enabled	4
FPP ×32	Disabled	Disabled	1
	Disabled	Enabled	4
	Enabled	Disabled	8
	Enabled	Enabled	8

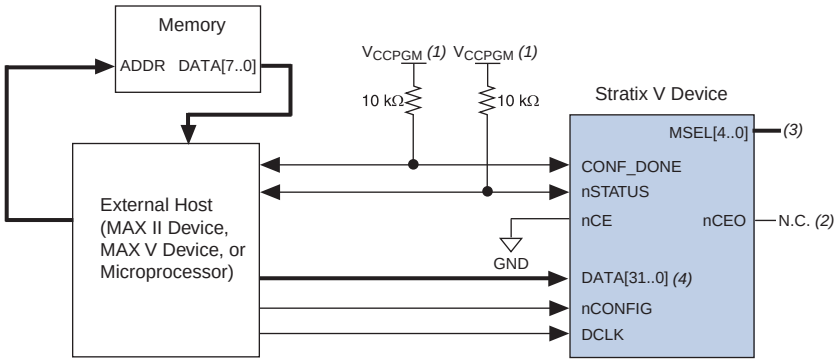
Note to Table 9–6:

(1) Depending on the DCLK-to-DATA[] ratio, the host must send a DCLK frequency that is r times the data rate in bytes per second (Bps), or words per second (Wps). For example, in FPP ×16 when the DCLK-to-DATA[] ratio is 2, the DCLK frequency must be 2 times the data rate in Wps. Stratix V devices use the additional clock cycles to decrypt and decompress the configuration data.

If the DCLK-to-DATA[] ratio is greater than 1, at the end of configuration, you can only stop the DCLK (DCLK-to-DATA[] ratio – 1) clock cycles after the last data is latched into the Stratix V device.

Figure 9–1 shows the configuration interface connections between the Stratix V device and a MAX II or MAX V device for single device configuration.

Figure 9–1. Single Device FPP Configuration Using an External Host



Notes to Figure 9–1:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different data width, configuration voltage standards, and POR delay. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) If you use FPP ×8, use DATA[7..0]. If you use FPP ×16, use DATA[15..0].

FPP Multi-Device Configuration

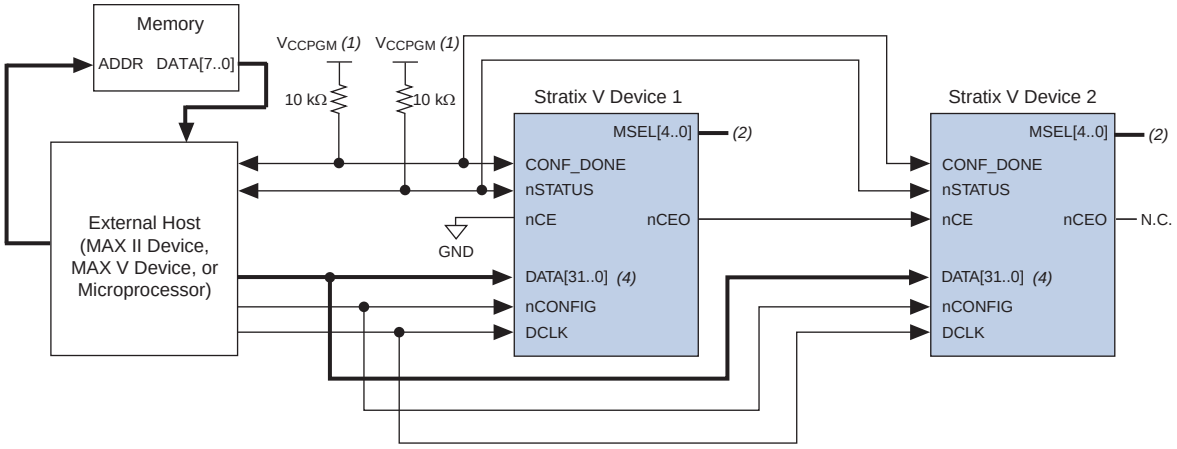
For FPP multi-device configuration, you can configure all devices with different sets of configuration data (multiple SRAM object files [**.sof**s]) or with the same configuration data (single **.sof**). In both cases, the nCONFIG, nSTATUS, DCLK, DATA[], and CONF_DONE pins are connected to every device in the chain. Ensure that the DCLK and data line are buffered for every fourth device. This ensures the signal integrity and prevents clock skew problems.

Because all device's CONF_DONE and nSTATUS pins are tied together, all devices initialize and enter user mode at the same time. If any device detects an error, configuration stops for the entire chain and you must reconfigure all devices. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

☞ For FPP multi-device configuration, all devices in the chain must have the same data width. If you are using FPP ×32, all devices in the chain must use FPP ×32 configuration scheme. If you are using FPP ×8, you can use the Stratix V device with other FPGA devices that support FPP ×8.

Figure 9–2 shows how to configure multiple devices using a MAX II or MAX V device when both devices receive a different set of configuration data (multiple **.sof**s). You must convert the multiple **.sof**s into a single programming file for the FPP chain programming.

Figure 9–2. Multi-Device FPP Configuration Using an External Host When Both Devices Receive a Different Set of Configuration Data



Notes to Figure 9–2:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different data width and POR delay. To connect MSEL, refer to Table 9–4 on page 9–7.

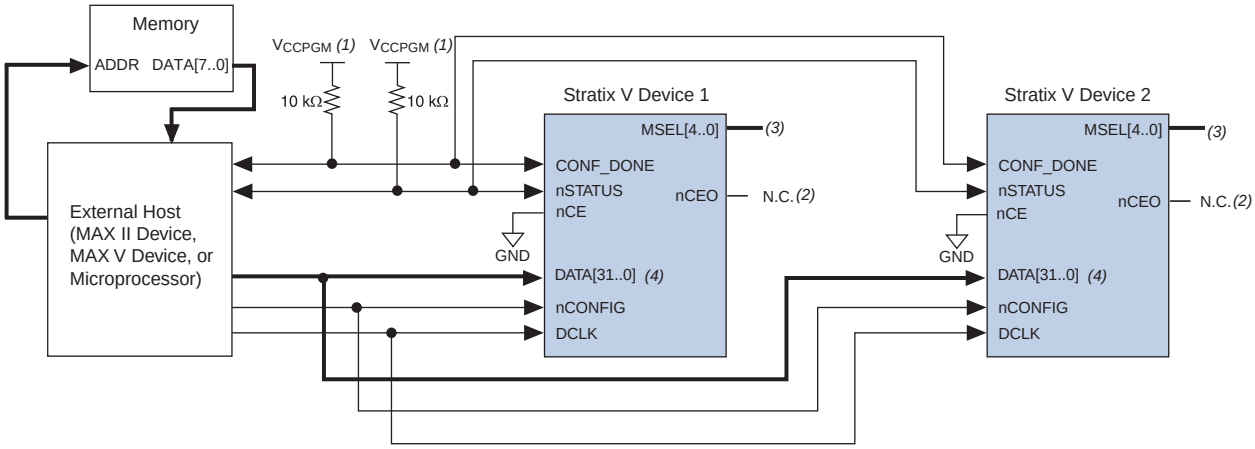
(4) If you use FPP ×8, use DATA[7..0]. If you use FPP ×16, use DATA[15..0]. All devices in the chain must have the same data width.

(5) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

In Figure 9–2, after the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data to the second device is transparent to the MAX II device, MAX V device, or microprocessor.

Figure 9–3 shows the FPP configuration setup for multiple devices when both Stratix V devices receive the same configuration data (single **.sof**).

Figure 9–3. Multiple Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 9–3:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different data width and POR delay. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) If you use FPP ×8, use DATA[7..0]. If you use FPP ×16, use DATA[15..0]. All devices in the chain must have the same data width.

(5) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

In Figure 9–3, because both nCE pins are tied to GND, both devices in the chain begin and complete the configuration and enter user mode at the same time.

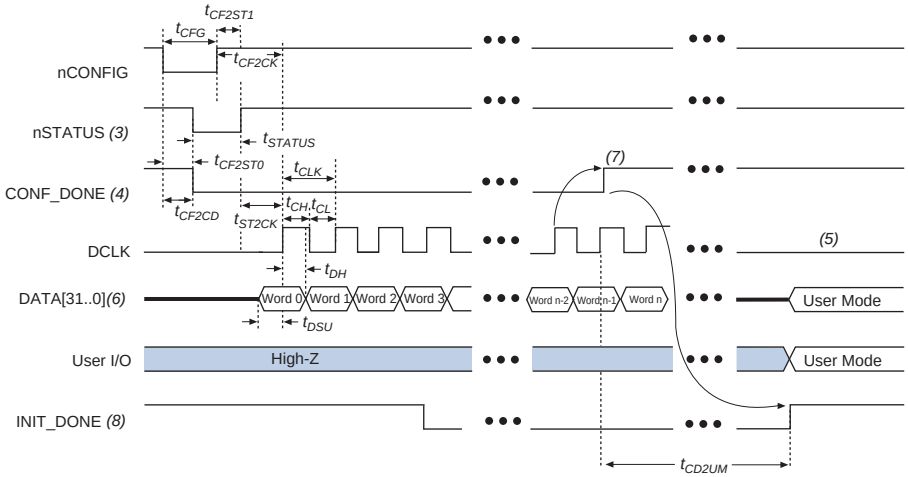
To configure FPP multi-device with a single **.sof**, all Stratix V devices in the chain must be in the same package and density.

FPP Configuration Timing

Figure 9–4 shows the timing waveform for FPP configuration when using a MAX II or MAX V device as an external host. This waveform shows timing when the DCLK-to-DATA[] ratio is 1.

When you enable the decompression or design security feature, the DCLK-to-DATA[] ratio varies for FPP ×8, FPP ×16, and FPP ×32. For the respective DCLK-to-DATA[] ratio, refer to Table 9–6 on page 9–9.

Figure 9–4. FPP Configuration Timing Waveform When the DCLK-to-DATA[] Ratio is 1 *(1), (2)*



Notes to Figure 9–4:

(1) Use this timing waveform when the DCLK-to-DATA[] ratio is 1.

(2) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic-high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(3) After power-up, the Stratix V device holds nSTATUS low for the time of the POR delay.

(4) After power-up, before and during configuration, CONF_DONE is low.

(5) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.

(6) For FPP ×16, use DATA[15..0]. For FPP ×8, use DATA[7..0]. DATA[31..0] are available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings.

(7) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high when the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(8) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–7 lists the timing parameters for Stratix V devices for FPP configuration when the DCLK-to-DATA[] ratio is 1.

Table 9–7. FPP Timing Parameters for Stratix V Devices *(1), (2)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(4)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns
t_{DH}	DATA[] hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	$0.45 \times 1/f_{MAX}$	—	s
t_{CL}	DCLK low time	$0.45 \times 1/f_{MAX}$	—	s
t_{CLK}	DCLK period	$1/f_{MAX}$	—	s
f_{MAX}	DCLK frequency (FPP ×8/×16)	—	125	MHz
	DCLK frequency (FPP ×32)	—	100	MHz
t_{CD2UM}	CONF_DONE high to user mode *(5)*	175	437	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (17,408 × CLKUSR period) *(6)*	—	—

Notes to Table 9–7:

(1) This information is preliminary.

(2) Use these timing parameters when the decompression and design security features are disabled.

(3) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

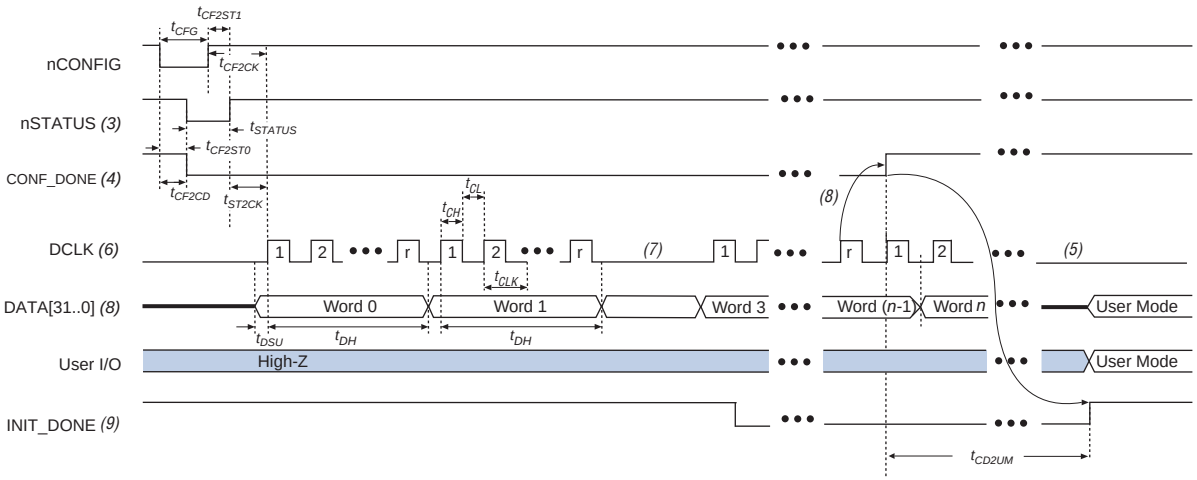
(4) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.

(5) The minimum and maximum numbers apply only if you chose the internal oscillator as the clock source for initializing the device.

(6) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

Figure 9–5 shows the timing waveform for FPP configuration when using a MAX II device, MAX V device, or microprocessor as an external host. This waveform shows timing when the DCLK-to-DATA[] ratio is more than 1.

Figure 9–5. FPP Configuration Timing Waveform When the DCLK-to-DATA[] Ratio is >1 *(1), (2)*



Notes to Figure 9–5:

(1) Use this timing waveform and parameters when the DCLK-to-DATA[] ratio is >1. To find out the DCLK-to-DATA[] ratio for your system, refer to Table 9–6 on page 9–9.

(2) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(3) After power-up, the Stratix V device holds nSTATUS low for the time as specified by the POR delay.

(4) After power-up, before and during configuration, CONF_DONE is low.

(5) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.

(6) "r" denotes the DCLK-to-DATA[] ratio. For the DCLK-to-DATA[] ratio based on the decompression and the design security feature enable settings, refer to Table 9–6 on page 9–9.

(7) If needed, pause DCLK by holding it low. When DCLK restarts, the external host must provide data on the DATA[31..0] pins prior to sending the first DCLK rising edge.

(8) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high after the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(9) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

If you are using the Stratix V decompression and design security features, configuration data is latched on the rising edge of every first DCLK cycle out of the N DCLK cycles where N is the DCLK-to-DATA ratio. Altera recommends keeping the data on DATA[31..0] stable for the next N–1 clock cycles when the data is being processed.

When you are using the decompression and design security features, you can only stop DCLK during N–1 clock cycles after the last data byte was latched into the Stratix V device.

Table 9–8 lists the timing parameters for Stratix V devices for FPP configuration when the DCLK-to-DATA[] ratio is more than 1.

Table 9–8. FPP Timing Parameters for Stratix V Devices When the DCLK-to-DATA[] Ratio is >1 *(1), (2)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns
t_{DH}	DATA[] hold time after rising edge on DCLK	N–1/f_{DCLK} *(6)*	—	s
t_{CH}	DCLK high time	0.45 × 1/f_{MAX}	—	s
t_{CL}	DCLK low time	0.45 × 1/f_{MAX}	—	s
t_{CLK}	DCLK period	1/f_{MAX}	—	s
f_{MAX}	DCLK frequency (FPP ×8/×16)	—	125	MHz
	DCLK frequency (FPP ×32)	—	100	MHz
t_R	Input rise time	—	40	ns
t_F	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	175	437	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (17,408 × CLKUSR period) *(5)*	—	—

Notes to Table 9–8:

(1) This information is preliminary.

(2) Use these timing parameters when you use the decompression and design security features.

(3) You can obtain this value if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(4) The minimum and maximum numbers apply only if you use the internal oscillator as the clock source for initializing the device.

(5) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

(6) N is the DCLK-to-DATA ratio and f_{DCLK} is the DCLK frequency the system is operating.

Active Serial Configuration (Serial Configuration Devices)

The AS configuration scheme supports AS ×1 (1-bit data width) and AS ×4 (4-bit data width) modes. Serial configuration device (EPCS) supports AS ×1 mode and quad-serial configuration device (EPCQ) supports AS ×1 and AS ×4 modes. The AS ×4 mode provides four times faster configuration time than the AS ×1 mode.

EPCS and EPCQ are low-cost devices with non-volatile memory that feature a simple four-pin interface or six-pin interface, respectively, and a small form factor. These features make the AS configuration scheme an ideal low-cost configuration solution.

If you wish to gain control of the EPCS pins, hold the nCONFIG pin low and pull the nCE pin high. This causes the device to reset and tri-state the AS configuration pins.

For more information about EPCS and EPCQ devices, refer to the volume 2 of the *Configuration Handbook*.

AS mode supports DCLK frequency up to 100 MHz. You can choose the CLKUSR or internal oscillator as the configuration clock source that drives DCLK. If you use the internal oscillator as the configuration clock source, you can choose a 12.5, 25, 50, or 100 MHz clock from the **Configuration** panel in the **Device and Pins Option** settings.

Table 9–9 lists the DCLK frequency specification in the AS configuration scheme.

Table 9–9. DCLK Frequency Specification in the AS Configuration Scheme *(1), (2), (3)*

Minimum	Typical	Maximum	Unit
5.3	7.9	12.5	MHz
10.6	15.7	25.0	MHz
21.3	31.4	50.0	MHz
42.6	62.9	100.0	MHz

Notes to Table 9–9:

(1) This information is preliminary.

(2) This applies to the DCLK frequency specification when using the internal oscillator as the configuration clock source.

(3) The AS multi-device configuration scheme does not support DCLK frequency of 100 MHz.

You can choose the internal oscillator or CLKUSR as the DCLK clock source by selecting the option under **Device and Pins Option** settings, **Configuration** panel in the Quartus II software. This sets specific option in the programming file. By default, in AS scheme, Stratix V devices power-up and begin configuration with 12.5 MHz internal oscillator as the DCLK clock source. After reading the option bits from the programming file, Stratix V devices continue using the internal oscillator at 12.5 MHz frequency, switch to a higher internal oscillator clock frequency, or switch to the CLKUSR pin.

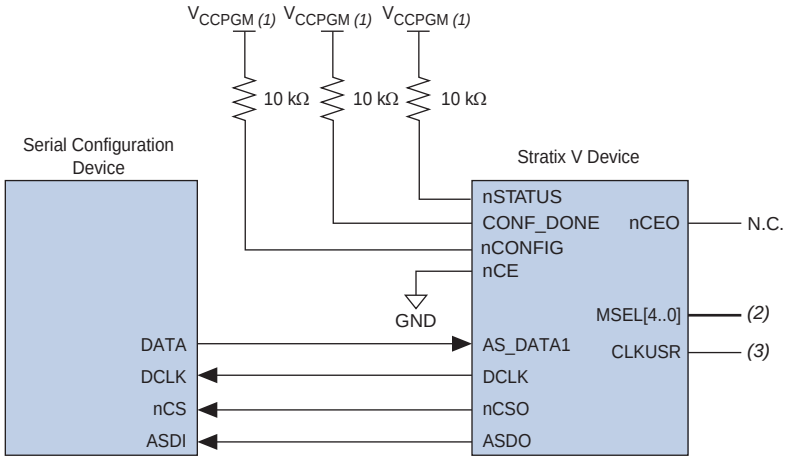
If you choose CLKUSR as configuration clock source, the maximum frequency allowed is 100 MHz.

During device configuration, Stratix V devices read the configuration data using the serial interface, decompress the data if necessary, and configure their SRAM cells. In the AS scheme, the Stratix V device controls the configuration interface. In the PS scheme, the external host (a MAX II device, MAX V device, or microprocessor) controls the interface.

You can select between the AS ×1 and AS ×4 settings by selecting the option under **Device and Pins Option** settings, **Configuration** panel in the Quartus II software. This sets a specific option bit in the programming file. By default, in the AS scheme, Stratix V devices power-up and begin configuration as an AS ×1 mode. Upon reading the option bits from the programming file, Stratix V devices either stay as an AS ×1 mode or switch to an AS ×4 mode for the rest of the configuration.

Figure 9–6 shows the single-device configuration setup for an AS ×1 mode.

Figure 9–6. Single Device AS ×1 Mode Configuration



Notes to Figure 9–6:

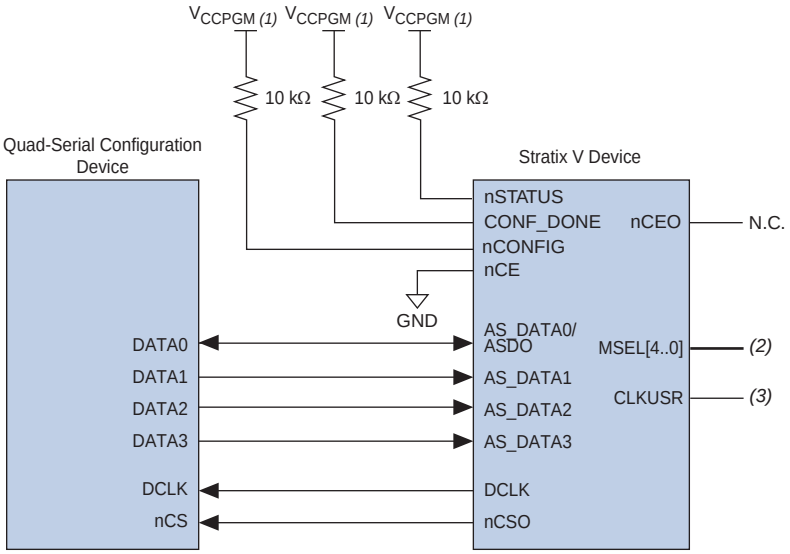
(1) Connect the pull-up resistors to V_{CCPGM} at 3.0-V supply.

(2) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(3) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–7 shows the single-device configuration setup for an AS ×4 mode.

Figure 9–7. Single Device AS ×4 Mode Configuration



Notes to Figure 9–7:

(1) Connect the pull-up resistors to V_{CCPGM} at 3.0-V supply.

(2) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(3) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

The serial clock (DCLK) generated by the Stratix V device controls the entire configuration cycle and provides timing for the serial interface. Stratix V devices use an internal oscillator or external clock (CLKUSR) as the configuration clock source (DCLK). In the AS configuration scheme, Stratix V devices drive out control signals on the falling edge of DCLK and latch in the data on the following falling edge of DCLK.

During configuration, Stratix V devices enable the EPCS or EPCQ by driving the nCSO output pin low, which connects to the chip select (nCS) pin of the EPCS or EPCQ. Stratix V devices use the serial clock (DCLK) and serial data output (ASDO) pins to send operation commands and read address signals to the EPCS or EPCQ. The EPCS or EPCQ provides data on its serial data output (DATA[]) pin, which connects to the AS_DATA[] input of the Stratix V devices.

AS Multi-Device Configuration

For the AS multi-device configuration scheme, you can configure all devices with different sets of configuration data (different **.sof**) or with the same configuration data (same **.sof**). In both cases, the nCONFIG, nSTATUS, DCLK, and data line (AS_DATA1 on the master device and DATA0 on the slave device) and CONF_DONE pins are connected to every device in the chain. Ensure that the DCLK and data line are buffered for every fourth device.

☞ The AS configuration scheme supports multi-device in AS ×1 mode. The AS ×4 mode does not support multi-device configuration setup.

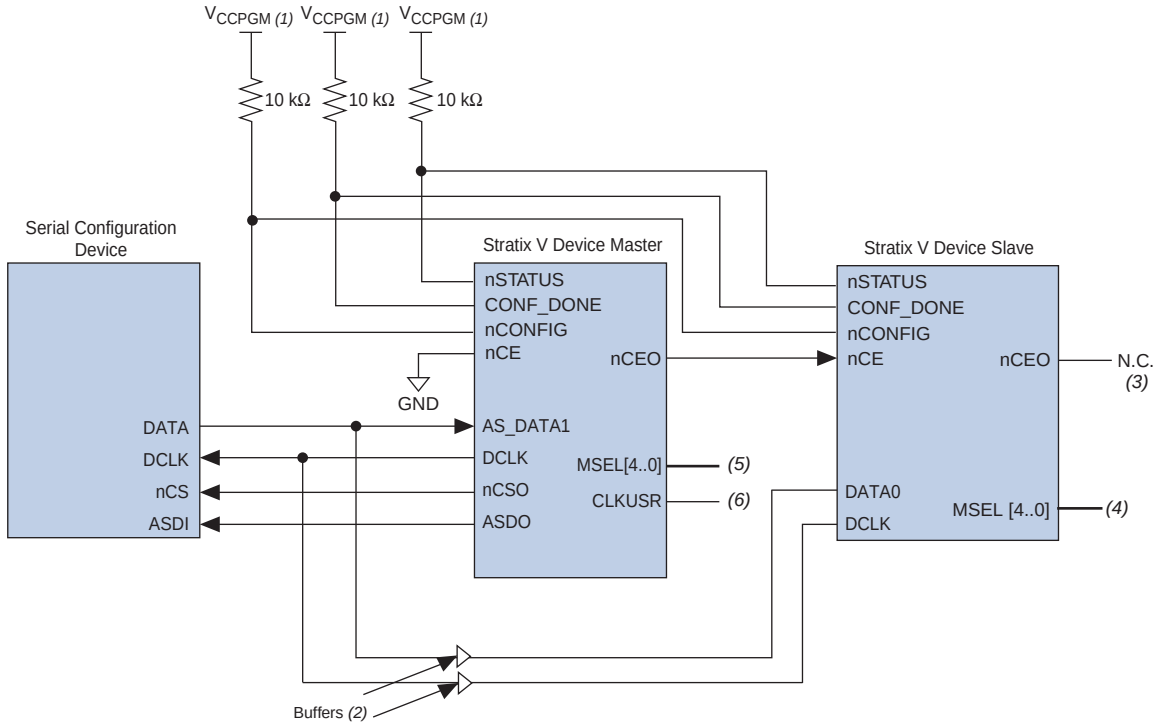
☞ The AS multi-device configuration scheme does not support DCLK frequency of 100 MHz.

In the AS multi-device configuration, the nSTATUS, nCONFIG, and CONF_DONE pins are tied together. Therefore, all devices initialize and enter user mode at the same time. If any device detects an error, configuration stops for the entire chain and you must reconfigure all devices. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

☞ In this configuration scheme, the first Stratix V device in the chain is the configuration master and controls the configuration of the entire chain. You must connect its MSEL pins to select the AS configuration scheme. The remaining Stratix V devices are configuration slaves. You must connect their MSEL pins to select the PS configuration scheme. Any other Altera® device that supports a PS configuration can also be part of the chain as the configuration slave.

Figure 9–8 shows the multi-device configuration setup for AS ×1 mode when both devices in the chain receive different sets of configuration data (multiple **.sof**s).

Figure 9–8. AS Multi-Device Configuration When Both Devices in the Chain Receive Different Sets of Configuration Data *(1), (2)*



Notes to Figure 9–8:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.

(2) Connect the repeater buffers between the Stratix V master and slave device for AS_DATA1/DATA0 and DCLK.

(3) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(4) For the appropriate MSEL settings based on POR delay settings, set the slave device MSEL setting to the PS scheme. Refer to Table 9–4 on page 9–7.

(5) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

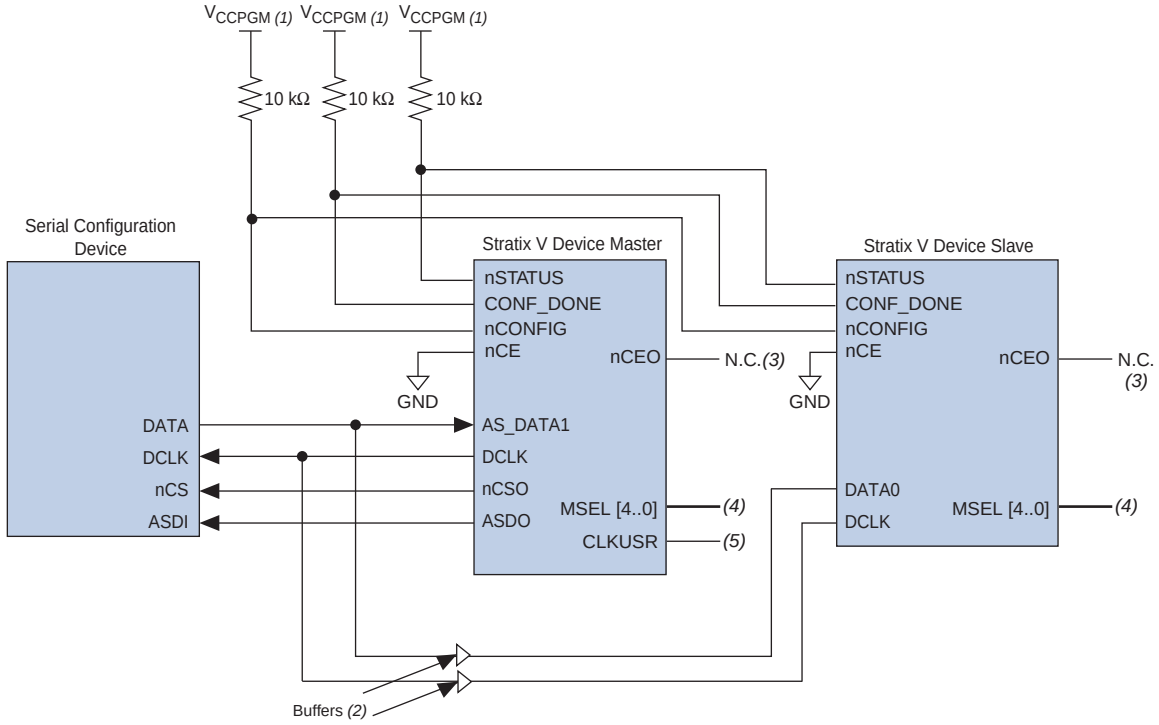
(6) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration.

(7) For 50 MHz frequency, delay DCLK in reference to DATA0 by a minimum of 5 ns and a maximum of 10 ns.

In Figure 9–8, after the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data to the second device is transparent to the first device in the chain.

Figure 9–9 shows the multi-device configuration setup for AS ×1 mode when all devices in the chain receive the same set of configuration data (single **.sof**).

Figure 9–9. AS Multi-Device Configuration When the Devices Receive the Same Data Using a Single .sof *(1)*



Notes to Figure 9–9:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.
(2) Connect the repeater buffers between the Stratix V master and slave device for AS_DATA1/DATA0 and DCLK.
(3) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.
(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.
(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration.
(6) For 50 MHz frequency, delay DCLK in reference to DATA0 by a minimum of 5 ns and a maximum of 10 ns.

AS Connection Guidelines

Table 9–10 lists the board trace length and loading between the supported EPCS or EPCQ and Stratix V devices for single- and multi-device AS configurations.

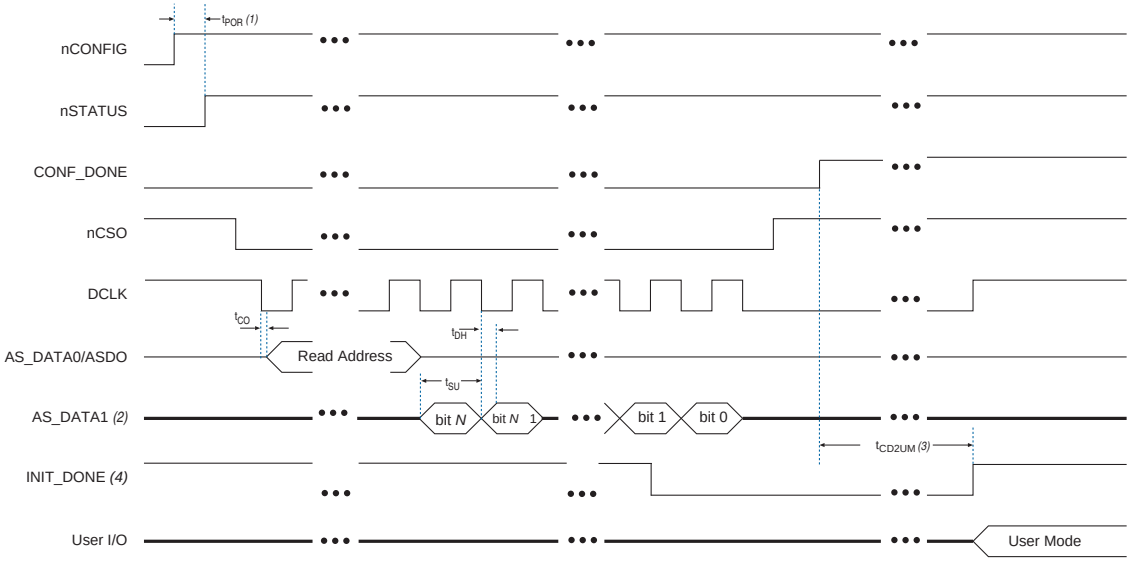
Table 9–10. Maximum Trace Length and Loading for AS ×1/×4 Configurations

Stratix V Device AS Pins	Maximum Board Trace Length from the Stratix V Device to the Serial Configuration Device for 12.5/ 25/ 50 MHz Operation (Inches)	Maximum Board Trace Length from the Stratix V Device to the Serial Configuration Device for 100 MHz Operation (Inches)	Maximum Board Load (pF)
DCLK	10	6	5
DATA[3..0]	10	6	10
nCSO	10	6	10

AS Configuration Timing

Figure 9–10 shows the timing waveform for the AS ×1 mode and AS ×4 mode configuration timing.

Figure 9–10. AS Configuration Timing



Notes to Figure 9–10:

(1) The AS scheme supports standard and fast POR delay (t_{POR}). For t_{POR} delay information, refer to "POR Delay Specification" on page 9–2.
(2) If you are using AS ×4 mode, this signal represents the AS_DATA[3..0] and EPCQ sends in 4-bits of data for each DCLK cycle.
(3) The initialization clock can be from internal oscillator or CLKUSR pin.
(4) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–11 lists the timing parameters for AS ×1 and AS ×4 configurations in Stratix V devices.

Table 9–11. AS Timing Parameters for AS ×1 and AS ×4 Configurations in Stratix V Devices *(1), (2), (3)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CO}	DCLK falling edge to AS_DATA0/ASDO output	—	4	µs
t_{SU}	Data setup time before falling edge on DCLK	1.5	—	ns
t_{H}	Data hold time after falling edge on DCLK	0	—	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	175	437	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (17,408 × CLKUSR period)	—	—

Notes to Table 9–11:

(1) This information is preliminary.

(2) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for initializing the device.

(3) t_{CF2CD}, t_{CF2ST0}, t_{CFG}, t_{STATUS}, and t_{CF2ST1} timing parameters are identical to the timing parameters for PS mode listed in Table 9–12 on page 9–32.

(4) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on this pin, refer to "Initialization" on page 9–5.

Estimating the Active Serial Configuration Time

The AS configuration time is dominated by the time it takes to transfer data from the EPCS to the Stratix V device. This serial interface is clocked by the Stratix V DCLK.

You can estimate the minimum AS ×1 mode configuration time by using the following equation:

.rbf Size × (minimum DCLK period / 1 bit per DCLK cycle) = estimated minimum configuration time.

You can estimate the minimum AS ×4 mode configuration time by using the following equation:

.rbf Size × (minimum DCLK period / 4 bits per DCLK cycle) = estimated minimum configuration time.

Enabling compression reduces the amount of configuration data that is transmitted to the Stratix V device, which also reduces the configuration time. Your configuration time is reduced based on the compression ratio. The compression ratio varies based on the design.

Programming EPCS and EPCQ

EPCS and EPCQ are non-volatile, flash-memory-based devices. You can program these devices in-system using a USB-Blaster™, EthernetBlaster, or ByteBlaster™ II download cable. Alternatively, you can program EPCS or EPCQ device using a microprocessor with the SRunner software driver.

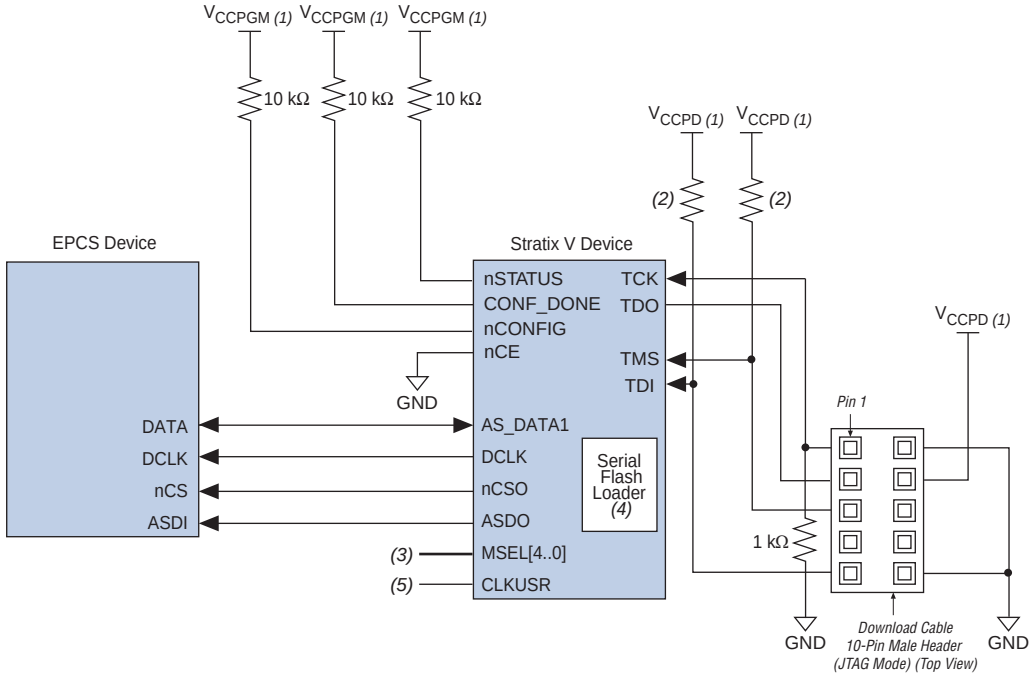
If you are not using Quartus II software or SRunner software for EPCQ 256 programming, put your EPCQ 256 device into four-byte addressing mode before you program and configure your device.

For more information about SRunner software driver, refer to *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming*.

In-system programming offers you an option to program the EPCS or EPCQ device either using an AS programming interface or a JTAG interface. Using the AS programming interface, the configuration data is programmed into the EPCS by the Quartus II software or any supported third-party software. Using the JTAG interface, an Altera IP called Serial Flash Loader (SFL) must be downloaded into the Stratix V device to form a bridge between the JTAG interface and the EPCS or EPCQ device. This allows the EPCS or the EPCQ device to be programmed directly using the JTAG interface.

Figure 9–11 shows the connection setup when programming the EPCS device using the JTAG interface.

Figure 9–11. Connection Setup for Programming the EPCS Device Using the JTAG Interface



Notes to Figure 9–11:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V power supply. For more information about how to connect to V_{CCPD}, refer to "V_{CCPD} Pin" on page 9–3.

(2) Resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

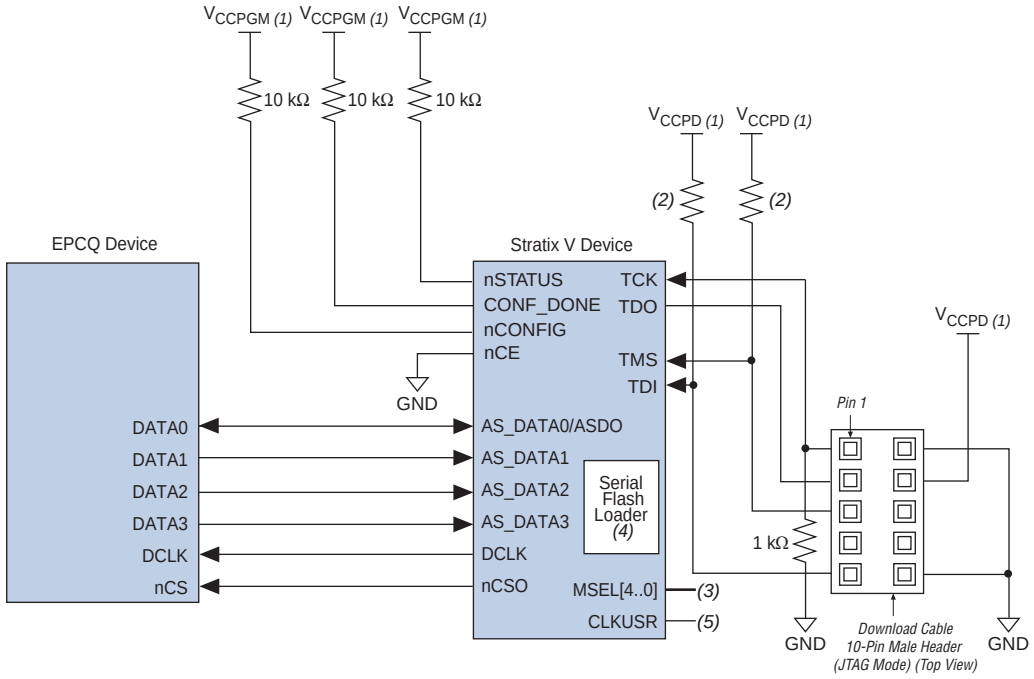
(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) Instantiate SFL in your design to form a bridge between the EPCS device and the Stratix V device. For more information about SFL, refer to *AN 370: Using the Serial Flash Loader with the Quartus II Software*.

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–12 shows the connection setup when programming the EPCQ device using the JTAG interface.

Figure 9–12. Connection Setup for Programming the EPCQ Device Using the JTAG Interface



Notes to Figure 9–12:

(1) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

(2) Resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

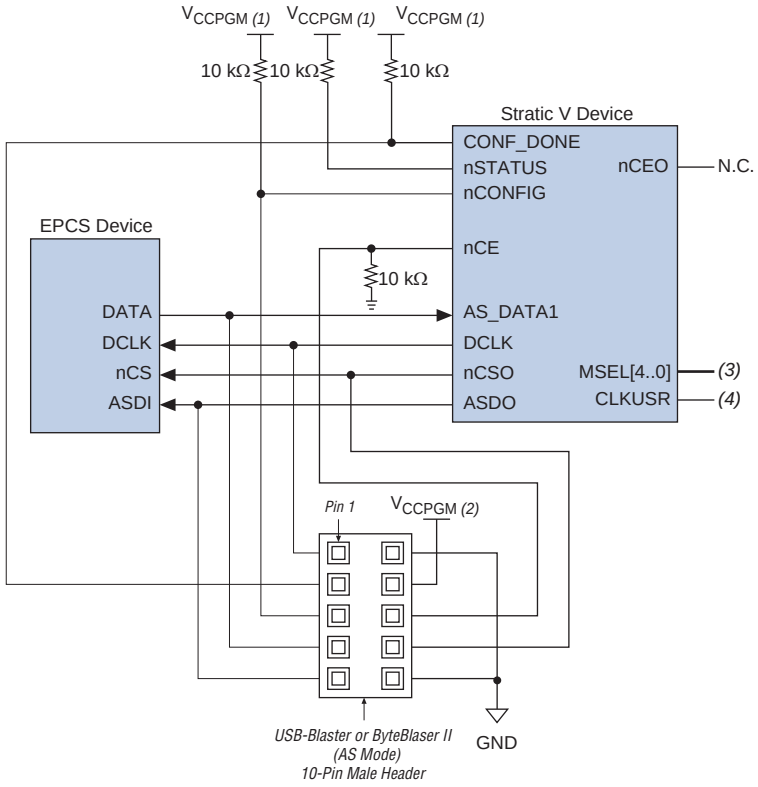
(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) Instantiate SFL in your design to form a bridge between the EPCS device and the Stratix V device. For more information about SFL, refer to *AN 370: Using the Serial Flash Loader with the Quartus II Software*.

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–13 shows the connection setup when programming the EPCS device using the AS interface.

Figure 9–13. Connection Setup for Programming the EPCS Device Using the AS Interface



Notes to Figure 9–13:

(1) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

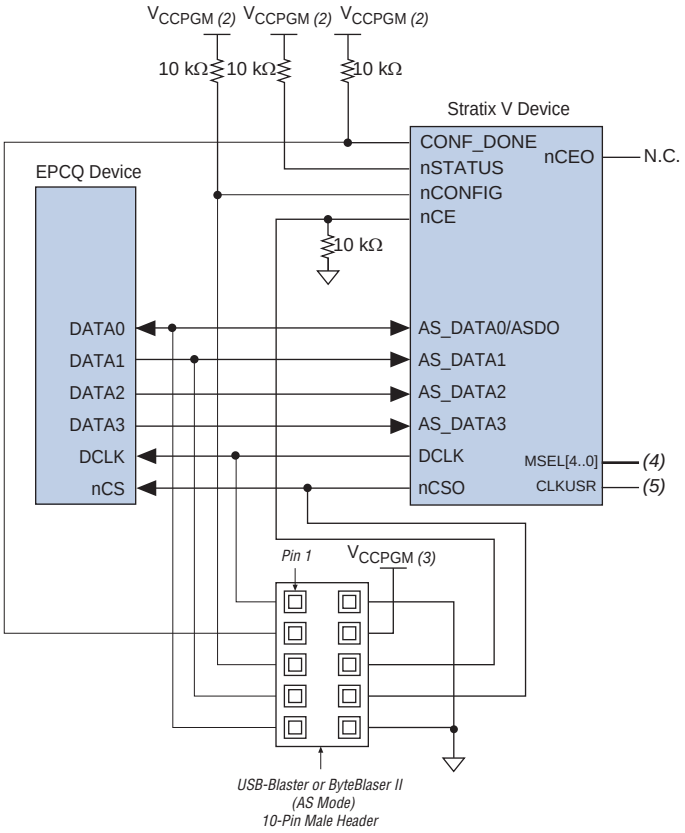
(2) Power up the USB-ByteBlaster, ByteBlaster II, or EthernetBlaster cable's $V_{CC(TRGT)}$ with V_{CCPGM}.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–14 shows the connection setup when programming the EPCQ device using the AS interface.

Figure 9–14. Connection Setup for Programming the EPCQ Device Using the AS Interface *(1)*



Notes to Figure 9–14:

(1) Using the AS header, the programmer transmits the operation commands and the configuration bits to the EPCQ device serially on DATA0.This is equivalent to the programming operation for the EPCS device as shown in Figure 9–14.

(2) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

(3) Power up the USB-ByteBlaster, ByteBlaster II, or EthernetBlaster cable's $V_{CC(TRGT)}$ with V_{CCPGM}.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

During EPCS and EPCQ programming, the download cable disables the device access to the AS interface by driving the nCE pin high. The nCONFIG line is also pulled low to hold the Stratix V device in reset stage. After programming completes, the download cable releases nCE and nCONFIG, allowing the pull-down and pull-up resistors to drive the pin to GND and V_{CCPGM}, respectively.

During EPCQ programming using the download cable, DATA0 carries the programming data, operation command, and address information from the download cable into the EPCQ device. During EPCQ verification using the download cable, DATA1 carries the programming data back to the download cable.

Passive Serial Configuration

You can perform PS configuration of Stratix V devices using an external host such as a MAX II device, MAX V device, microprocessor, or a host PC. Therefore, the design that controls the configuration stages, such as fetching the data from flash memory and sending it to the device, must be stored in the external host.

The Parallel Flash Loader (PFL) feature in MAX II or MAX V devices provide an efficient method to program CFI flash memory devices through the JTAG interface. PFL also acts as a controller to read configuration data from the flash memory device and configures the Stratix V device. PFL supports both the PS and FPP configuration schemes.

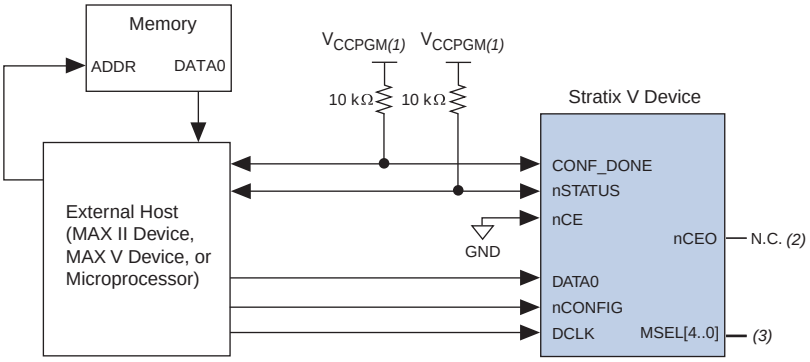
For more information about the PFL, refer to the *Parallel Flash Loader Megafunction User Guide*.

PS Configuration Using a MAX II Device, MAX V Device, or Microprocessor

The external host (a MAX II device, MAX V device, or microprocessor) reads configuration data from storage devices, such as flash memory, and transfers it to Stratix V devices. You can store the configuration data in **.pof**, **.rbf**, **.hex**, or **.ttf** format. If you are using configuration data in **.rbf**, **.hex**, or **.ttf** format, you must send the LSB of each data byte first. For example, if the **.rbf** file contains the byte sequence 02 1B EE 01 FA, the serial data transmitted to the device must be 0100-0000 1101-1000 0111-0111 1000-0000 0101-1111.

Figure 9–15 shows the configuration interface connections between a Stratix V device and a MAX II or MAX V device for single device configuration.

Figure 9–15. Single Device PS Configuration Using an External Host



Notes to Figure 9–15:

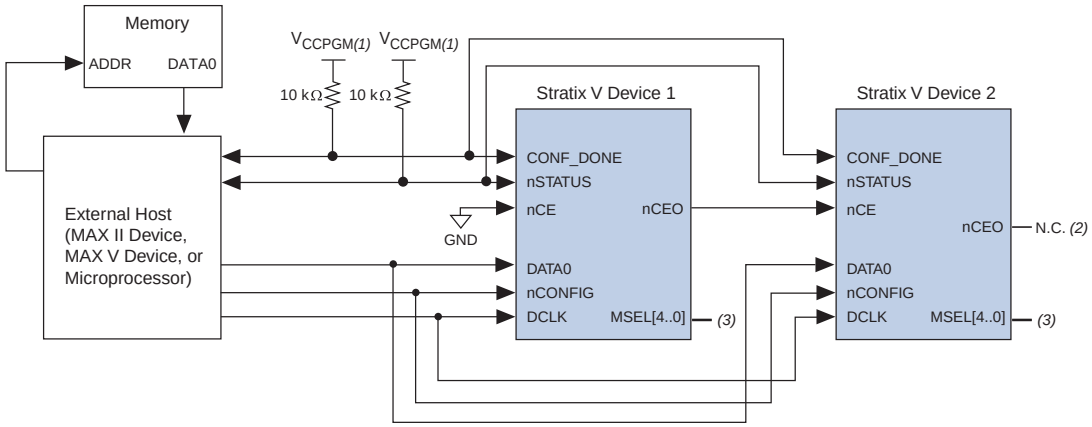
(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all the configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

Figure 9–16 shows the PS multi-device configuration using an external host when all devices in the chain receive different sets of configuration data (multiple **.sof**s).

Figure 9–16. PS Multi-device Configuration when Both Devices Receive Different Sets of Configuration Data



Notes to Figure 9–16:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all the configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

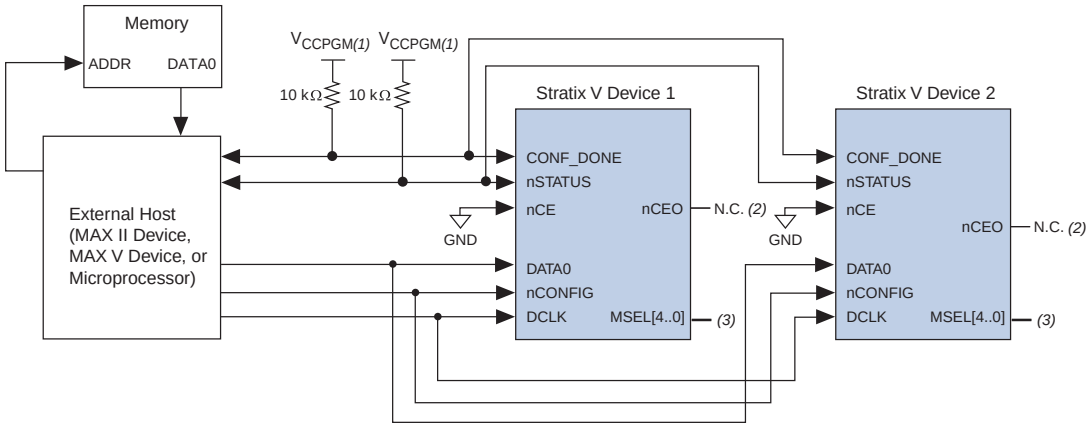
(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

In Figure 9–16, after the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data to the second device is transparent to the external host.

Figure 9–17 shows the PS multi-device configuration when all devices receive the same set of configuration data (single **.sof**).

Figure 9–17. PS Multi-device Configuration When Both Devices Receive the Same Set of Configuration Data



Notes to Figure 9–17:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all the configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

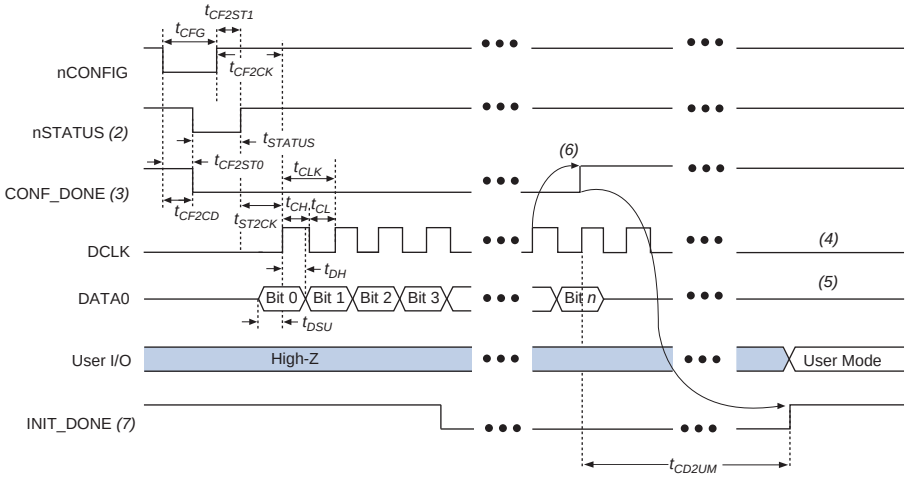
In Figure 9–17, because both nCE pins are tied to GND, both devices in the chain begin and complete the configuration and enter user mode at the same time.

To configure the PS multi-device with a single **.sof**, as shown in Figure 9–17, all Stratix V devices in the chain must be in the same package and density.

PS Configuration Timing

Figure 9–18 shows the timing waveform for a PS configuration when using a MAX II device, MAX V device, or microprocessor as an external host.

Figure 9–18. PS Configuration Timing Waveform *(1)*



Notes to Figure 9–18:

(1) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(2) After power-up, the Stratix V device holds nSTATUS low for the time of the POR delay.

(3) After power-up, before and during configuration, CONF_DONE is low.

(4) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.

(5) DATA0 is available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings in the **Device and Pins Option**.

(6) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high after the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(7) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–12 lists the PS configuration timing parameters for Stratix V devices.

Table 9–12. PS Timing Parameters for Stratix V Devices *(1)* **(Part 1 of 2)**

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(2)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns
t_{DH}	DATA[] hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	$0.45 \times 1/f_{MAX}$	—	s
t_{CL}	DCLK low time	$0.45 \times 1/f_{MAX}$	—	s

Table 9–12. PS Timing Parameters for Stratix V Devices (1) (Part 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t_{CLK}	DCLK period	$1/f_{MAX}$	—	s
f_{MAX}	DCLK frequency	—	125	MHz
t_{CD2UM}	CONF_DONE high to user mode (4)	175	437	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (17,408 × CLKUSR period) (5)	—	—

Notes to Table 9–12:

(1) This information is preliminary.
(2) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.
(3) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.
(4) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for initializing the device.
(5) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device for both uncompressed and compressed configuration data in the PS configuration scheme.

PS Configuration Using a Download Cable

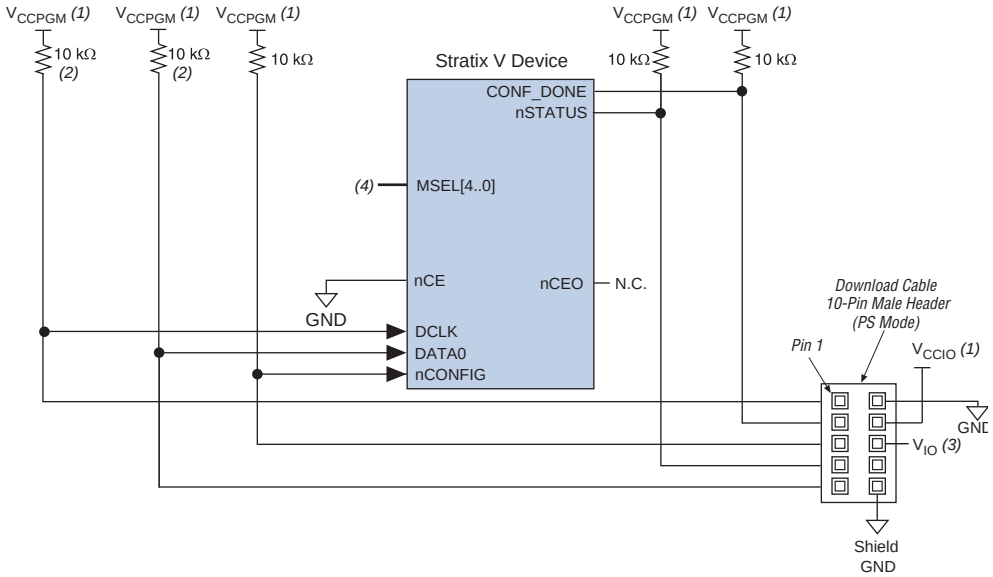
In this section, the generic term "download cable" includes the Altera USB-Blaster universal serial bus (USB) port download cable, ByteBlaster II parallel port download cable, and EthernetBlaster download cable.

In a PS configuration with a download cable, a PC acts as a host to transfer data from a storage device to the Stratix V device using the download cable. During configuration, the programming hardware or download cable places the configuration data one bit at a time on the device's DATA0 pin. The configuration data is clocked into the target device until CONF_DONE goes high.

If you turn on the CLKUSR option during PS configuration using a download cable and the Quartus II programmer, you do not have to provide a clock on the CLKUSR pin to initialize your device.

Figure 9–19 shows a PS configuration for Stratix V devices using an Altera download cable.

Figure 9–19. PS Configuration Using an Altera Download Cable



Notes to Figure 9–19:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCIO}) as the USB-Blaster, ByteBlaster II, or EthernetBlaster cable.

(2) You only need the pull-up resistors on DATA0 and DCLK if the download cable is the only configuration scheme used on your board. This ensures that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a MAX II device, MAX V device, or microprocessor, you do not need the pull-up resistors on DATA0 and DCLK.

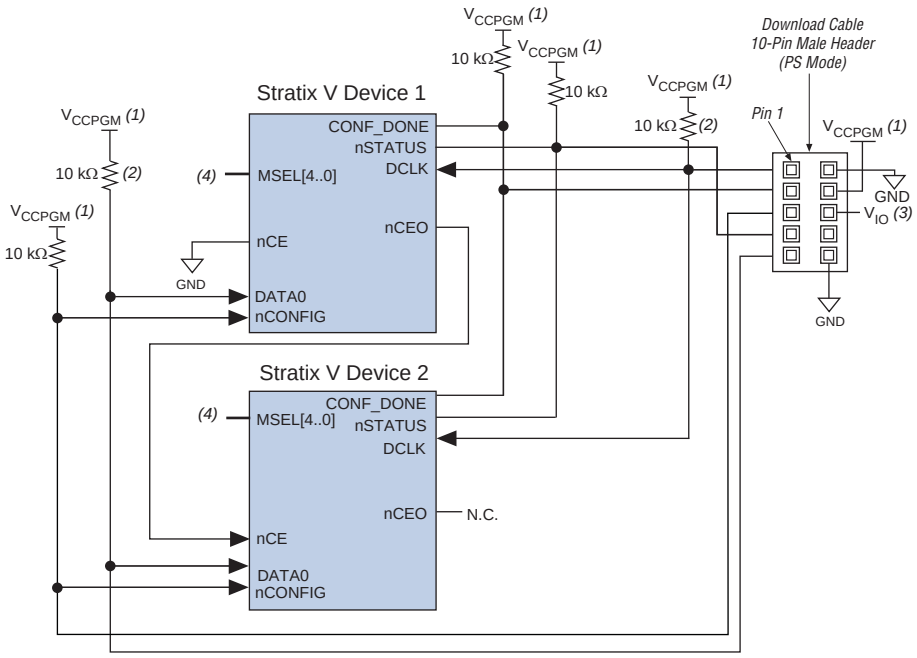
(3) In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when you use for AS programming; otherwise, it is a no connect.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL[4..0], refer to Table 9–4 on page 9–7.

Multi-Device PS Configuration Using Download Cable

You can use a download cable to configure multiple Stratix V devices as shown in Figure 9–20.

Figure 9–20. Multi-Device PS Configuration Using an Altera Download Cable



Notes to Figure 9–20:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCIO}) as the USB-Blaster, ByteBlaster II, or EthernetBlaster cable.

(2) You only need the pull-up resistors on DATA0 and DCLK if the download cable is the only configuration scheme used on your board. This ensures that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, you do not need the pull-up resistors on DATA0 and DCLK.

(3) In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when you use it for AS programming; otherwise, it is a no connect.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL[4..0], refer to Table 9–4 on page 9–7.

(5) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

In Figure 9–20, after the first device completes configuration, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE pins are connected to every device in the chain. As all the CONF_DONE and nSTATUS pins are tied together, all devices initialize and enter user mode at the same time. If any device detects an error, configuration stops for the entire chain and you must reconfigure all the devices. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

JTAG Configuration

You can use the same JTAG interface specifically developed for boundary-scan test (BST) to shift the configuration data into the device. The Quartus II software automatically generates a **.sof** that you can use for JTAG configuration with a download cable in the Quartus II software programmer.

For more information about JTAG BST and the commands available using Stratix V devices, refer to the following documents:

- *JTAG Boundary-Scan Testing in Stratix V Devices* chapter
- Programming Support for Jam STAPL Language

Stratix V devices are designed such that JTAG instructions have precedence over any device configuration modes. JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration of Stratix V devices during a PS configuration, the PS configuration is terminated and the JTAG configuration begins. All user I/O pins are tri-stated during the JTAG configuration.

You cannot use the Stratix V decompression or design security features if you are configuring your Stratix V device using JTAG-based configuration.

For more information about `TDI`, `TDO`, `TMS`, and `TCK`, refer to "Device Configuration Pins" on page 9–40.

For instructions to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing in Stratix V Devices* chapter.

To configure a single device in a JTAG chain, the programming software places all the other devices in bypass mode. In bypass mode, devices pass programming data from the `TDI` pin to the `TDO` pin through a single bypass register without being affected internally. This scheme enables the programming software to program or verify the target device. Configuration data driven into the device appears on the `TDO` pin one clock cycle later. The Quartus II software verifies successful JTAG configuration after completion by checking the state of `CONF_DONE` through the JTAG port.

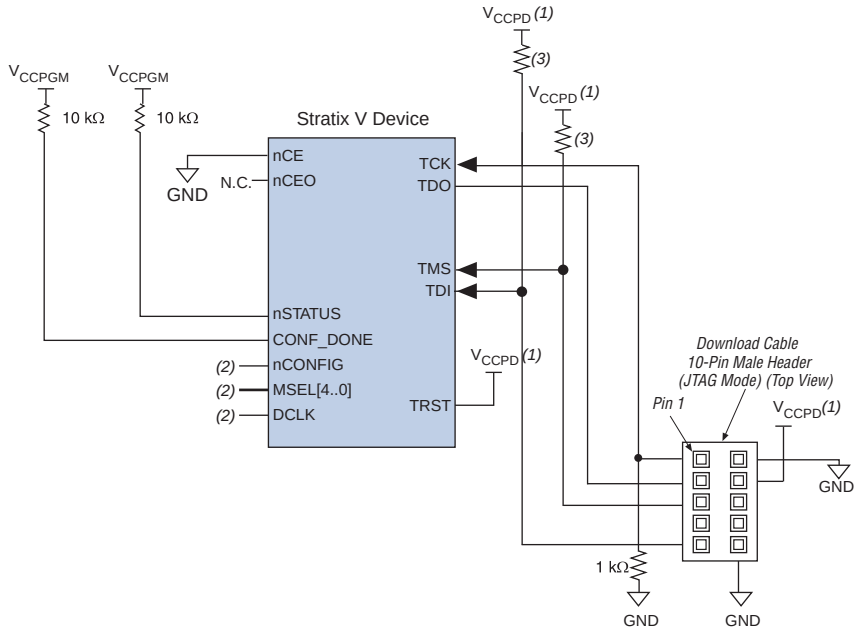
If `CONF_DONE` is not high, the Quartus II software indicates that configuration has failed. If `CONF_DONE` is high, the software indicates that configuration was successful. After the configuration data is transmitted serially using the JTAG `TDI` port, the `TCK` port is clocked an additional 1,222 cycles to perform device initialization.

The chip-wide reset (`DEV_CLRn`) and chip-wide output enable (`DEV_OE`) pins on Stratix V devices do not affect JTAG boundary-scan or programming operations.

You can generate a JAM File (**.jam**) or Jam-byte Code (**.jbc**) to be used with other third party programmer tools. Alternatively, you can use JRunner with **.rbf** to program your device.

Figure 9–21 shows the JTAG configuration of a single Stratix V device.

Figure 9–21. JTAG Configuration of a Single Device Using a Download Cable



Notes to Figure 9–21:

(1) Connect the pull-up resistor V_{CCPD}. For more information, refer to V_{CCPD} requirement in "V_{CCPD} Pin" on page 9–3.

(2) If you only use JTAG configuration, connect `nCONFIG` to V_{CCPGM} and `MSEL[4..0]` to GND. Pull `DCLK` either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, connect `MSEL[4..0]`, `nCONFIG` and `DCLK` based on the selected configuration scheme.

(3) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

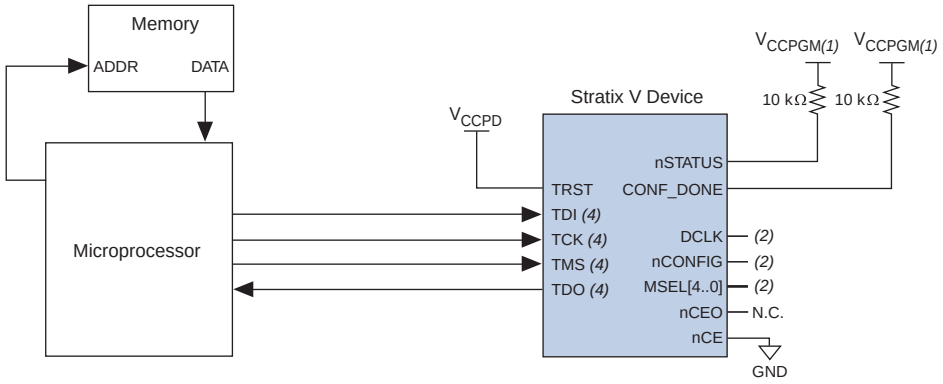
(4) You must connect `nCE` to GND or drive it low for successful JTAG configuration.

Alternatively, you can use a microprocessor to program the device through the JTAG interface. You can use JRunner as your software driver.

For more information about JRunner, refer to *AN 414: The JRunner Software Driver: An Embedded Solution for PLD JTAG Configuration*.

Figure 9–22 shows a JTAG configuration of a Stratix V device using a microprocessor.

Figure 9–22. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 9–22:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for all Stratix V devices in the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device.

(2) If you only use the JTAG configuration, connect nCONFIG to V_{CCPGM} and MSEL[4..0] to GND. Pull DCLK either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, set the MSEL[4..0] and tie nCONFIG and DCLK based on the selected configuration scheme.

(3) Connect nCE to GND or drive it low for successful JTAG configuration.

(4) The microprocessor must use the same I/O standard as V_{CCPD} to drive the JTAG pins.

(5) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

CONFIG_IO Instruction

The CONFIG_IO instruction allows you to configure I/O buffers using the JTAG port and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix V device or waiting for a configuration device to complete configuration. After configuration is interrupted and JTAG testing is complete, you must reconfigure the part using the JTAG interface or if you support FPP, PS, or AS on your board, reconfigure the device by externally pulsing nCONFIG low. Alternatively, you can pulse nCONFIG low through the same JTAG interface using the PULSE_NCONFIG JTAG instruction.

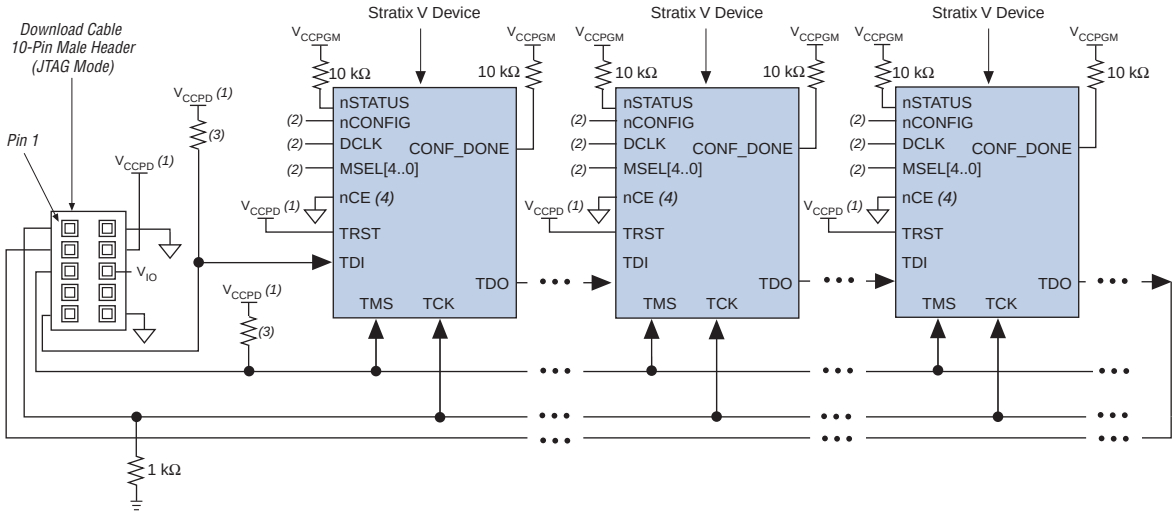
All JTAG instructions (except BYPASS, IDCODE, and SAMPLE) can be issued by first interrupting the configuration and reprogramming the I/O pins using the CONFIG_IO instruction.

Multi-Device JTAG Configuration

When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the TCK, TDI, and TMS pins with an on-board buffer. You can place other Altera devices that have JTAG support in the same JTAG chain for device programming.

JTAG-chain device programming is ideal when the system contains multiple devices or when testing your system using JTAG BST circuitry. Figure 9–23 shows a multi-device JTAG configuration.

Figure 9–23. JTAG Configuration of Multiple Devices Using a Download Cable



Notes to Figure 9–23:

(1) Connect the pull-up resistor V_{CCPD}. For more information, refer to V_{CCPD} requirement in "V_{CCPD} Pin" on page 9–3.

(2) If you only use JTAG configuration, connect `nCONFIG` to V_{CCPGM} and `MSEL[4..0]` to GND. Pull `DCLK` either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, connect the `MSEL[4..0]`, `nCONFIG`, and `DCLK` based on the selected configuration scheme.

(3) The resistor value can vary from 1kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

(4) You must connect `nCE` to GND or drive it low for successful JTAG configuration.

If you want to use the JTAG multi-device configuration in conjunction with other schemes, such as a FPP, PS, or AS, tie `CONF_DONE`, `nSTATUS`, and `nCONFIG` together as recommended in the FPP, PS, or AS multi-device configuration schemes. Ensure that the JTAG chain is the same order as the multi-device FPP, PS, or AS configuration chain.

If you only use JTAG configuration, Altera recommends connecting the circuitry as shown in Figure 9–22, where each of the `CONF_DONE` and `nSTATUS` signals are isolated to enable each device to enter user mode individually.

For more information about combining the JTAG configuration with other configuration schemes, refer to the *Combining Different Configuration Schemes* chapter in volume 2 of the *Configuration Handbook*.

For more information about JTAG and Jam STAPL in embedded environments, refer to *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*. To download the Jam player, visit the Altera website.

For more information about how to use the USB-Blaster, ByteBlaster II, or EthernetBlaster cables, refer to the following user guides:

- *USB-Blaster Download Cable User Guide*
- *ByteBlaster II Download Cable User Guide*
- *Ethernet Blaster Communications Cable User Guide*

Device Configuration Pins

Table 9–13 and Table 9–14 list the connections and functionality of all the configuration-related pins on the Stratix V devices. Table 9–13 lists the Stratix V configuration pins and their power supply.

Table 9–13. Configuration Pin Summary for Stratix V Devices (Part 1 of 2)

Description	Input/Output	User Mode	Powered By	Configuration Scheme
TDI	Input	—	V_{CCPD}	JTAG
TMS	Input	—	V_{CCPD}	JTAG
TCK	Input	—	V_{CCPD}	JTAG
TRST	Input	—	V_{CCPD}	JTAG
TDO	Output	—	V_{CCPD}	JTAG
CLKUSR	Input	I/O (1)	V_{CCPGM}/V_{CCIO} (4)	All schemes
CRC_ERROR	Output	I/O (1)	Pull-up	Optional, all schemes
CONF_DONE	Bidirectional	—	V_{CCPGM}/Pull-up	All schemes
DATA0	Bidirectional	I/O (2)	V_{CCPGM}/V_{CCIO} (4)	FPP, PS
DATA[31..1]	Bidirectional	I/O (2)	V_{CCPGM}/V_{CCIO} (4)	FPP
DCLK	Input	—	V_{CCPGM}	FPP, PS
	Output	—	V_{CCPGM}	AS
DEV_OE	Input	I/O (1)	V_{CCPGM}/V_{CCIO} (4)	Optional, all schemes
DEV_CLRn	Input	I/O (1)	V_{CCPGM}/V_{CCIO} (4)	Optional, all schemes
INIT_DONE	Output	I/O (1)	Pull-up	Optional, all schemes
MSEL[4..0]	Input	—	V_{CCPGM}	All schemes
nSTATUS	Bidirectional	—	V_{CCPGM}	All schemes
nCE	Input	—	V_{CCPGM}	All schemes
nCEO	Output	I/O (3)	Pull-up	All schemes
nCONFIG	Input	—	V_{CCPGM}	All schemes
nCSO	Output	—	V_{CCPGM}	AS
nIO_PULLUP	Input	—	V_{CC} (5)	All schemes
AS_DATA0/ASDO	Bidirectional	—	V_{CCPGM}	AS

Table 9–13. Configuration Pin Summary for Stratix V Devices (Part 2 of 2)

Description	Input/Output	User Mode	Powered By	Configuration Scheme
AS_DATA[3..1]	Bidirectional	—	V_{CCPGM}	AS

Notes to Table 9–13:

(1) This is a dual-purpose pin. This pin is available as an I/O if the associated option that enables this pin is turned off from the **Configuration** panel in the **Device and Pins Option** settings. For example, the DEV_OE is available as a user I/O if the **Enable device-wide output enable** option is turned off.

(2) This is a dual-purpose pin. The state of this pin in user mode depends on the **Dual-purpose Pins** settings in the **Device and Pins Option**.

(3) This pin is available as an I/O if this pin is not feeding the next device's nCE in a multi-device configuration. To use this pin to feed the next device's nCE in a multi-device chain, turn on **Enable INIT_DONE output** option under **Device and Pins Option**, **General** panel in the Quartus II Software.

(4) This pin is powered up by V_{CCPGM} during configuration. It is powered up by V_{CCIO} of the bank in which the pin resides if it is used as a regular I/O in user mode.

(5) Although nIO_PULLUP is powered up by V_{CC}, Altera recommends connecting this pin to V_{CCPGM} or GND directly without using a pull-up or pull-down resistor.

Table 9–14 lists the configuration pin descriptions.

Table 9–14. Configuration Pins Description (Part 1 of 3)

Pin Name	Description
TDI (1)	Dedicated test data input. Serial input pin for instructions as well as test and programming data. Data is shifted on the rising edge of TCK. This pin has an internal 25-kΩ pull-up that is always active.
TMS (1)	Dedicated test mode select. Input pin that provides the control signal to determine the transitions of the TAP controller state machine. TMS is evaluated on the rising edge of TCK. Therefore, you must set up TMS before the rising edge of TCK. Transitions in the state machine occur on the falling edge of TCK after the signal is applied to TMS. This pin has an internal 25-kΩ pull-up that is always active.
TCK (1)	Dedicated test clock input. Clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. It is expected that the clock input waveform have a nominal 50% duty cycle. This pin has an internal 25-kΩ pull-down that is always active.
TRST (1)	Dedicated test reset input. Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to the IEEE Std. 1149.1 standard. Connecting this pin low disables the JTAG circuitry. This pin has an internal 25-kΩ pull-up that is always active.
TDO (1)	Dedicated test data output. Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. This pin is tri-stated if the data is not being shifted out of the device.
CLKUSR	Optional user-supplied clock input. It synchronizes the initialization of one or more devices. Enable this pin by turning on the **Enable user-supplied start-up clock (CLKUSR)** option under **Device and Pins Option**, **Configuration** panel in the Quartus II software.

Table 9–14. Configuration Pins Description (Part 2 of 3)

Pin Name	Description
CRC_ERROR	Optional output pin. Signals that the device has detected a cyclical redundancy check (CRC) error during user mode operation. this pin is an open-drain output pin by default and requires a 10 kΩ pull-up resistor. To use this pin as regular output, turn-off the **Enable Open-drain on CRC_ERROR pin** in **Device and Pins Option**, **Error Detection CRC** panel in the Quartus II software. The target device drives this pin low if there is no CRC error in user mode operation. As an open-drain output, if a CRC error occurs, the device releases the pin which is then pulled high by the external pull-up resistor. Enable this pin by turning on **Enable CRC error detection on CRC_ERROR pin** option in the Quartus II software. For more information about the CRC_ERROR pin, refer to *SEU Mitigation in Stratix V Devices* chapter.
CONF_DONE	Dedicated open-drain bidirectional pin. The target device drives the CONF_DONE pin low before and during configuration. After all the configuration data is received without error and the initialization cycle starts, the target device releases the CONF_DONE pin, which is then pulled high by the external pull-up resistor. The target device then reads the CONF_DONE pin status to ensure that the CONF_DONE is at logic high. After it is sensed high, the target device initializes and enters user mode. Driving CONF_DONE low after initialization completes does not affect the configured device.
DATA0 *(3)*	Dual-purpose data input pin. The data received on DATA0 is synchronized to DCLK. After configuration completes, this pin is available as a user I/O pin.
DATA[31..1] *(3)*	Dual-purpose data input pins. If you are using FPP ×16 or FPP ×32, only a subset of these pins are required for configuration. The pins that are not required for configuration, you can use them as regular I/Os. During configuration, byte-wide, or word-wide data is received on these pins. The data received on DATA[31..1] are synchronized to the DCLK.
DCLK	Dedicated bidirectional clock pin. In the PS and FPP configurations, DCLK is the clock input used to clock data from an external source into the target device. Data is latched into the device on the rising edge of DCLK. After configuration completes, drive DCLK high or low, whichever is more convenient. In the AS mode, DCLK is an output clock to clock the EPCS or EPCQ devices. Data is latched into the device on the falling edge of the DCLK. After AS configuration completes, this pin is tri-stated with a weak pull-up resistor. Toggling this pin after configuration does not affect the configured device.
DEV_OE *(2)*	Optional input pin that allows you to override all tri-states on the device. When this pin is driven low, all the I/O pins are tri-stated. When this pin is driven high, all the I/O pins behave as programmed. Enable this pin by turning on the **Enable device-wide output enable (DEV_OE)** option in the Quartus II software.
DEV_CLRn *(2)*	Optional input pin that allows you to override all clears on all the device registers. When this pin is driven low, all the registers are cleared. When this pin is driven high, all the registers behave as programmed. This pin is enabled by turning on the **Enable device-wide reset (DEV_CLRn)** option in the Quartus II software.
INIT_DONE *(2)*	Optional output pin. Signals when the device has initialized and is in user mode. During the reset stage, after the device exits POR, and during the beginning of the configuration, the INIT_DONE pin is tri-stated and pulled high due to an external pull-up resistor. After the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization completes, the INIT_DONE pin is released and pulled high and the device enters user mode. Thus, the monitoring circuitry must be able to detect a low-to-high transition. Enable this pin by turning on the **Enable INIT_DONE output** option in the Quartus II software.

Table 9–14. Configuration Pins Description (Part 3 of 3)

Pin Name	Description
MSEL[4..0]	Dedicated input pins. Five-bit configuration input that sets the Stratix V device configuration scheme. For the appropriate connections, refer to Table 9–4 on page 9–7. The MSEL[4..0] pins have internal 25-kΩ pull-down resistors that are always active.
nSTATUS	Dedicated open-drain bidirectional pin. The device drives nSTATUS low immediately after power-up and releases it after the device exits POR. During user mode and regular configuration, this pin is pulled high by an external 10-kΩ resistor. During configuration, the device drives this pin low to indicate an error during configuration. If an external source drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. This mechanism is used during multi-device configuration setup. If one of the devices in the chain has an error and pulls its nSTATUS pin low, it resets the entire chain. Driving nSTATUS low after configuration and initialization completes does not affect the configured device.
nCE	Dedicated active-low chip enable input pin. Driving this pin low allows configuration. Drives the nCE pin low during configuration, initialization, and user mode for all single-device configurations. For a multi-device configuration, connect the nCE pin to GND or to the nCEO of the previous device in the chain based on the recommendation in the respective configuration setup diagram.
nCEO *(3)*	Dual-purpose open-drain output pin. This pin drives low when device configuration completes. To use this pin to feed the next device's nCE pin in a multi-device chain, turn on **Enable INIT_DONE output** option under **Device and Pins Option**, **General** panel in the Quartus II Software. In a single-device configuration, this pin can be used as a regular I/O. In multi-device configuration, if this pin is not feeding the nCE of the next device, you can use it as a regular I/O.
nCONFIG	Dedicated input pin. A low pulse on this pin during configuration and user mode causes the device to enter a reset state and tri-states all the I/O pins. A low-to-high logic starts a reconfiguration. During JTAG programming, the nCONFIG status is ignored.
nCSO	Dedicated output pin. Drives the control signal from the Stratix V device to the EPCS and EPCQ devices in AS mode. After AS configuration completes, these pins are tri-stated with a weak pull-up resistor.
nIO_PULLUP	Dedicated input pin. This input pin enables or disables the internal pull-up resistors on the user I/O pins and dual purpose I/O pins (DATA[31..0], CLKUSR, INIT_DONE, DEV_OE, and DEV_CLRN). A logic high turns off the weak internal pull-up resistors, while a logic low turns them on. This pin has an internal 25-kΩ pull-down resistor that is always active.
AS_DATA0/ASDO	Dedicated bidirectional data pin. In AS ×1 and AS ×4 configurations, ASDO is used to send the operation command and addresses to the EPCS or EPCQ devices. During an AS ×4 configuration, the data is received on an AS_DATA0 and is synchronized to DCLK. After AS configuration completes, this pin is tri-stated with a weak pull-up resistor.
AS_DATA[3..1]	Dedicated bidirectional data pins. During an AS configuration, the data is received on these pins and is synchronized to DCLK. After AS configuration completes, these pins are tri-stated with a weak pull-up resistor.

Notes to Table 9–14:

(1) If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to logic high. For instructions to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing* chapter.

(2) This is a dual-purpose pin. This pin is available as an I/O if the associated option that enables this pin is turned off from the **Configuration** panel in the **Device and Pins Option** settings. For example, the DEV_OE is available as a user I/O if the **Enable device-wide output enable** option is turned off.

(3) This is a dual-purpose pin. The state of this pin in the user mode depends on **Dual-purpose Pins** settings in the **Device and Pins Option** settings.

Configuration Data Decompression

Stratix V devices support configuration data decompression, which saves configuration memory space and may shorten the configuration time. This feature allows you to store compressed configuration data in the configuration or other memory devices and transmit this compressed data to the Stratix V devices. During configuration, the Stratix V device decompresses the data in real time and programs its SRAM cells. The data decompression is done on-the-fly during configuration and does not require an additional processing time.

Preliminary data indicates that compression typically reduces the configuration data size by 30 to 55% based on the designs used. This reduces the storage requirement capacity for the flash memory. The decompression feature is supported in all configuration schemes except JTAG.

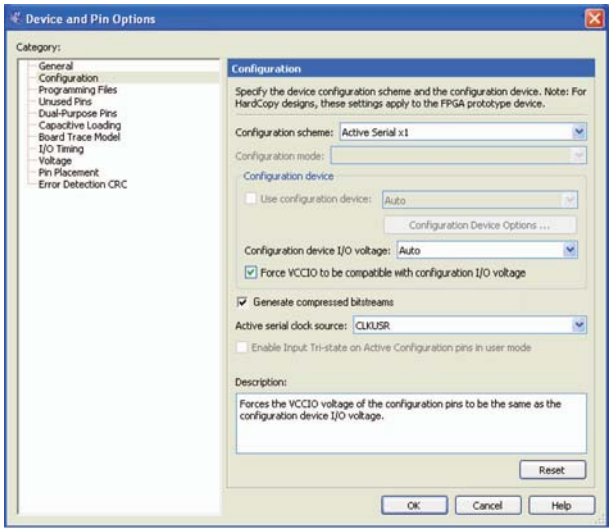
In FPP, enabling the decompression feature requires a different DCLK-to-DATA[] ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

There are two ways to enable compression for Stratix V data—before design compilation (in the Compiler Settings menu) and after design compilation (in the Convert Programming Files window).

To enable compression in the project's Compiler Settings menu, follow these steps:

1. On the Assignments menu, click **Device** to bring up the **Settings** dialog box.
2. After selecting your Stratix V device, open the **Device and Pin Options** dialog box.
3. In the **Configuration settings** panel, turn on the **Generate compressed bitstreams** option (refer to Figure 9–24).

Figure 9–24. Enabling Compression Bitstreams in Compiler Settings for Stratix V



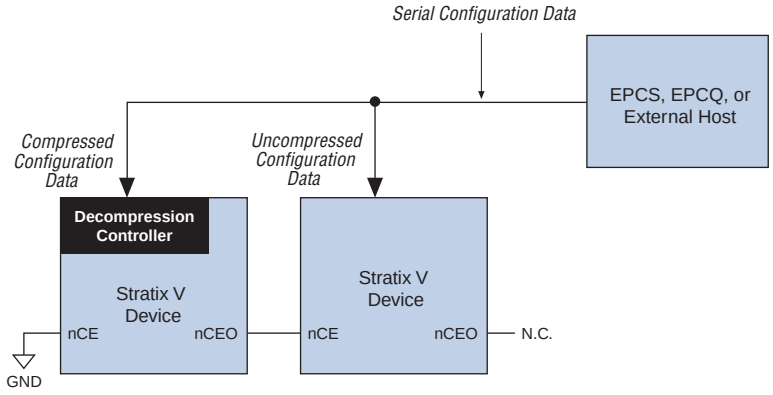
You can also enable compression when creating programming files from the **Convert Programming Files** window. To do this, follow these steps:

1. On the File menu, click **Convert Programming Files**.
2. Select the programming file type (**.pof**, **.sram**, **.hex**, **.hexout**, **.rbf**, or **.ttf**).
3. For POF output files, select a configuration device.
4. In the **Input files to convert** box, select **SOF Data**.
5. Select **Add File** and add a Stratix V device **.sof**.
6. Select the name of the file you added to the **SOF Data** area and click **Properties**.
7. Check the **Compression** check box.

If you are using a serial configuration scheme, AS ×1 or PS, for multi-device configuration, you can selectively enable the compression feature for each device in the chain. Figure 9–25 shows a chain of two Stratix V devices. The first Stratix V device has compression enabled and therefore receives compressed data from the external host. The second Stratix V device has the compression feature disabled and receives uncompressed data.

For FPP configuration schemes, a combination of compressed and uncompressed configuration in the same multi-device chain is not allowed due to the difference of the DCLK-to-DATA[] ratio.

Figure 9–25. Compressed and Uncompressed Serial Configuration Data in the Same Configuration File *(1)*



Note to Figure 9–25:

(1) The configuration for this setup can be generated from the Convert Programming Files menu in the Quartus II software.

Remote System Upgrades

This section describes the functionality and implementation of the dedicated remote system upgrade circuitry. It also defines several concepts related to remote system upgrades, including factory configuration, application configuration, remote update mode, and user watchdog timer. Additionally, this section provides design guidelines for implementing remote system upgrades with the supported configuration schemes.

System designers sometimes face challenges such as shortened design cycles, evolving standards, and system deployments in remote locations. Stratix V devices help overcome these challenges with their inherent reprogrammability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduce time-to-market, extend product life, and help to avoid system downtime.

Stratix V devices feature dedicated remote system upgrade circuitry. A soft logic (either the Nios® II embedded processor or user logic) implemented in a Stratix V device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to start a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, recovers from any error condition by reverting back to a safe configuration image, and provides error status information.

Remote system upgrades are supported in AS configuration schemes with EPCS and EPCQ devices. You can also implement remote system upgrades in conjunction with advanced Stratix V features such as real-time decompression of configuration data and design security using the advanced design security standard (AES) for secure and efficient field upgrades. The largest EPCS and EPCQ device currently supports 128 Mbits and 256 Mbits configuration data respectively.

Remote system upgrades are supported only in single-device configurations.

The remote system upgrade process in Stratix V devices involves the following steps:

1. A Nios II processor (or user logic) implemented in the Stratix V device logic array receives new configuration data from a remote location. The connection to the remote source uses a communication protocol such as TCP/IP, PCI, user datagram protocol (UDP), UART, or a proprietary interface.
2. The Nios II processor (or user logic) stores this new configuration data in non-volatile configuration memory.
3. The Nios II processor (or user logic) starts a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any errors that might occur during or after the reconfiguration cycle and provides error status information to the user design.

Figure 9–26 shows these remote system upgrade steps.

Figure 9–26. Functional Diagram of the Stratix V Remote System Upgrade Process

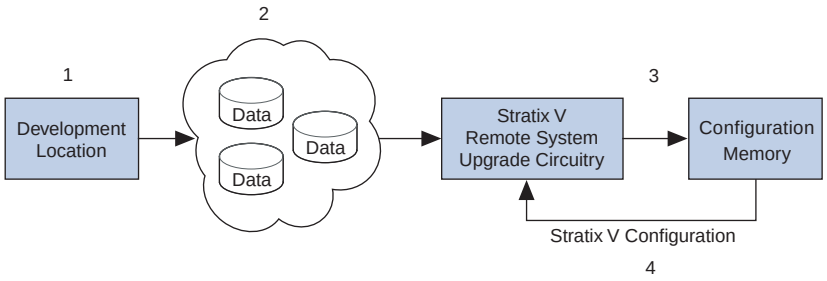
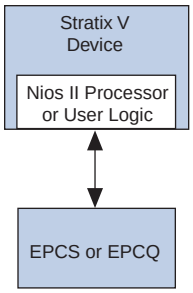


Figure 9–27 shows a block diagram for implementing a remote system upgrade with the Stratix V AS configuration scheme.

Figure 9–27. Remote System Upgrade Block Diagram for Stratix V AS Configuration Scheme *(1)*



Note to Figure 9–27:

(1) You must set the mode select pins (`MSEL[4..0]`) to **AS mode** to use remote system upgrade in your system. The `MSEL` pin settings vary for different POR delays. To connect `MSEL[4..0]`, refer to Table 9–4 on page 9–7.

Configuration Image Types

When performing a remote system upgrade, Stratix V device configuration data are classified as factory configuration images or application configuration images. An image, also referred to as a configuration image, is a design loaded into the Stratix V device that performs certain user-defined functions.

The factory image is a user-defined fall-back, or safe configuration, and is responsible for initiating the reconfiguration to a new image with the dedicated circuitry. Application images implement user-defined functionality in the target Stratix V device. You may also include the default application image functionality in the factory image. Each Stratix V device in your system requires one factory image and one or more application images.

Remote Update Mode

Stratix V remote system upgrade circuitry only supports remote update mode. In remote update mode, Stratix V devices load the factory configuration image after power up. The user-defined factory configuration determines which application configuration is to be loaded and triggers a reconfiguration cycle.

When a Stratix V device is first powered up in remote update mode, it loads the factory configuration located at the start address of `PGM[23..0]` = 24'h000000 in the EPCS and EPCQ devices. You must store your factory configuration image at this start address.

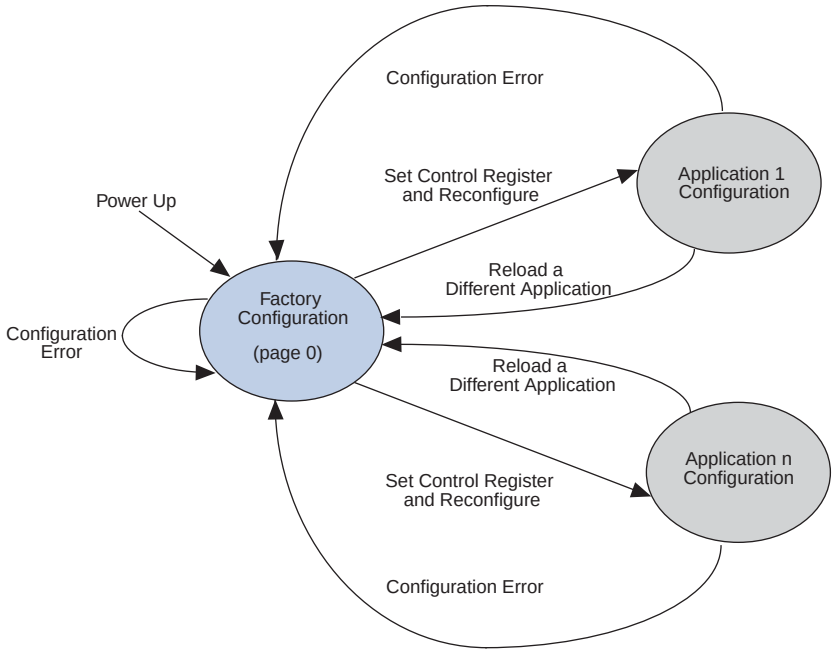
The application image start address can be at any EPCS or EPCQ sector boundary. Altera recommends using different sectors in the EPCS device for two images.

The factory image is user-designed and contains soft logic to perform the following:

- Process any errors based on status information from the dedicated remote system upgrade circuitry
- Communicate with the remote host and receive new application configurations and store this new configuration data in the local non-volatile memory device
- Determine which application configuration is to be loaded into the Stratix V device
- Enable or disable the user watchdog timer and load its time-out value
- Instruct the dedicated remote system upgrade circuitry to start a reconfiguration cycle

Figure 9–28 shows the transitions between the factory and application configurations in remote update mode.

Figure 9–28. Transitions Between Configurations in Remote Update Mode



After power up or a configuration error, the factory configuration image is loaded automatically. The system then decides to switch to the application configuration image or to stay in the factory configuration image. After the system decides to switch to an application configuration image, a reconfiguration is initiated through the remote system upgrade circuitry. In the application configuration image, the system may revert back to factory configuration image after the following reconfiguration trigger conditions are met:

- nSTATUS driven low externally
- Configuration CRC error
- User watchdog timer time-out
- Core nCONFIG signal assertion
- External nCONFIG signal assertion

After the factory configuration image is re-loaded, the user-designed factory configuration can read the remote system upgrade status register to determine the reason for the reconfiguration. The factory configuration then takes the appropriate error recovery steps and writes to the remote system upgrade control register to determine the next application configuration to be loaded.

Whenever the application configuration image is successfully loaded, the soft logic (Nios II processor or state machine and the remote communication interface) determine when the remote system update is arriving. When this occurs, the soft logic receives the incoming data, writes it to the configuration memory device, and triggers the device to load the factory configuration. The factory configuration reads the remote system upgrade status register and control register, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and initiates system reconfiguration.

Remote System Upgrade Using EPCQ 256

When you are using EPCQ 256, ensure that the application image address granularity is 32'h00000100. The **.rbf** size for your application image is 76,500 bytes longer than the numbers listed in Table 9–5 on page 9–8. You need to take this extra space requirement into consideration when you try to fit multiple application images in the EPCQ 256 device.

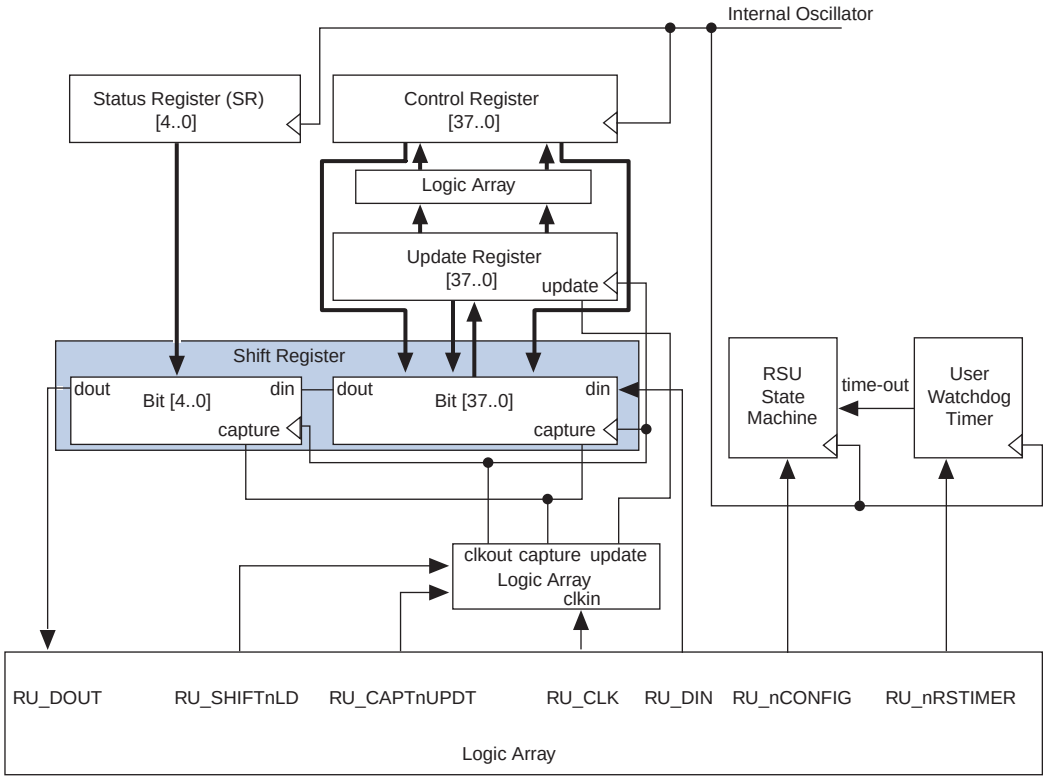
If you are not using Quartus II software or SRunner software for EPCQ 256 programming, put your EPCQ 256 device into four-byte addressing mode before you program and configure your device.

Dedicated Remote System Upgrade Circuitry

This section describes the implementation of the Stratix V dedicated remote system upgrades circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces with the user-defined factory and application configurations implemented in the Stratix V device logic array to provide the complete remote configuration solution. The remote system upgrade circuitry contains the remote system upgrade registers, a watchdog timer, and a state machine that controls those components.

Figure 9–29 shows the remote system upgrade circuitry.

Figure 9–29. Remote System Upgrade Circuitry *(1)*



Note to Figure 9–29:

(1) If you are using the ALTREMOTE_UPDATE megafunction, the RU_DOUT, RU_SHIFTnLD, RU_CAPTnUPDT, RU_CLK, RU_DIN, RU_nCONFIG, and RU_nRSTIMER signals are internally controlled by the megafunction to perform all the related remote system upgrade operations.

Table 9–15 lists the timing parameter specifications for the remote system upgrade circuitry.

Table 9–15. Remote System Upgrade Circuitry Timing Specifications

Parameter	Minimum	Maximum	Unit
$f_{MAX\_RU\_CLK}$ *(1)*	—	40	MHz
$t_{RU\_nCONFIG}$ *(2)*	250	—	ns
$t_{RU\_nRSTIMER}$ *(3)*	250	—	ns

Notes to Table 9–15:

(1) This clock is user-supplied to the remote system upgrade circuitry. If you are using the ALTREMOTE_UPDATE megafunction, the clock user-supplied to the ALTREMOTE_UPDATE megafunction must meet this specification.

(2) This is equivalent to strobing the reconfiguration input of the ALTREMOTE_UPDATE megafunction high for the minimum timing specification. For more information, refer to "Remote System Upgrade State Machine" on page 9–53.

(3) This is equivalent to strobing the reset_timer input of the ALTREMOTE_UPDATE megafunction high for the minimum timing specification. For more information, refer to "User Watchdog Timer" on page 9–53.

Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that store the page addresses, watchdog timer settings, and status information. Table 9–16 lists these registers.

Table 9–16. Remote System Upgrade Registers

Register	Description
Shift register	This register is accessible by the logic array and allows the update, status, and control registers to be written and sampled by user logic.
Control register	This register contains the current page address, user watchdog timer settings, and one bit specifying whether the current configuration is a factory configuration or an application configuration. During a read operation in an application configuration, this register is read into the shift register. When a reconfiguration cycle is initiated, the contents of the update register are written into the control register.
Update register	This register contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a capture in a factory configuration, this register is read into the shift register.
Status register	This register is written to by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register.

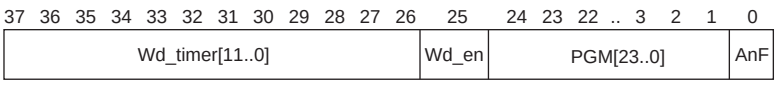
The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (RU_CLK).

Control Register

The control register stores the application configuration page address and user watchdog timer settings. The control register functionality depends on the remote system upgrade mode selection. A factory configuration in remote update mode has write access to this register.

Figure 9–30 shows the control register bit positions. Table 9–17 lists the control register bits. In the figure, the numbers show the bit position of a setting in a register. For example, bit number 25 is the enable bit for the watchdog timer.

Figure 9–30. Remote System Upgrade Control Register



The application-not-factory (AnF) bit indicates whether the current configuration loaded in the Stratix V device is the factory configuration or an application configuration. This bit is set low by the remote system upgrade circuitry when an error condition causes a fall-back to the factory configuration. When the AnF bit is high, the control register restricts the access to only read operations and enables the watchdog timer. The factory configuration design must set this bit high (1'b1) when updating the contents of the update register with the application page address and watchdog timer settings.

Table 9–17 lists the remote system upgrade control register contents.

Table 9–17. Remote System Upgrade Control Register Contents

Control Register Bit	Value (2)	Definition
AnF (1)	1'b0	Application not factory
PGM[23..0]	24'b0×000000	AS configuration start address (StAdd[23..0])
Wd_en	1'b0	User watchdog timer enable bit
Wd_timer[11..0]	12'b000000000000	User watchdog time-out value (most significant 12 bits of 29-bit count value: {Wd_timer[11..0], 17'b0})

Notes to Table 9–17:

(1) Factory configuration designs must set the AnF bit to **1'b1** before triggering the reconfiguration to application configuration image.

(2) This is the default value of the control register bit after the device exits POR and during reconfiguration back to the factory configuration image after reconfiguration trigger conditions.

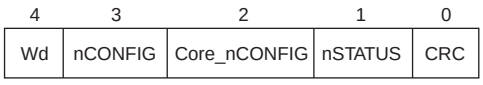
Status Register

The status register specifies the reconfiguration trigger condition. Figure 9–31 shows the status register content. The following list defines each bit:

- Bit 0—CRC error during application configuration
- Bit 1—nSTATUS assertion by an external device due to an error
- Bit 2—Stratix V device logic array triggered a reconfiguration cycle, possibly after downloading a new application configuration image
- Bit 3—external configuration reset (nCONFIG) assertion
- Bit 4—user watchdog timer time-out

Figure 9–31 shows the contents of the status register. The numbers in the figure show the bit positions in a 5-bit register.

Figure 9–31. Remote System Upgrade Status Register (1)



Note to Figure 9–31:

(1) After the device exits POR and powers-up, the status register content is 5'b00000.

Remote System Upgrade State Machine

After power-up, the shift register, control register, and update registers are reset to the values listed in Table 9–16, also known as POR reset values before the factory configuration image is loaded. In the factory configuration image, the user logic writes the AnF bit, page address, and watchdog timer settings for the next application configuration image to the update register. When the logic array configuration reset (RU_nCONFIG) goes low, the remote system upgrade state machine updates the control register with the contents of the update register, and triggers a reconfiguration to the new application configuration image.

If there is an error during reconfiguration to the new application configuration image, the remote system upgrade state machine directs the system to re-load a factory configuration image. The control and update registers are reset to POR reset values and the status register is updated with the error information. For example, if there is a CRC error during application configuration image configuration, the status register is updated with 5'b00001.

If there is no error during reconfiguration and the application configuration image is successfully loaded, the system stays in the application configuration image until another reconfiguration trigger condition occurs. This can be a core nCONFIG assertion, external nCONFIG assertion, or the watchdog timer time-out error. If this happens, the control register and update registers are reset to POR reset values and the status register is updated with the error information. Consequently, the system proceeds to load the factory configuration image. Based on the status register content, the user logic in the factory configuration image then decides to stay in the factory configuration image or reload a new application reconfiguration image.

Read operations during factory configuration access the contents of the update register. This feature is used by the factory configuration image user logic to verify that the page address and watchdog timer settings are written correctly. Read operations in application configurations access the contents of the control register. This information is used by the user logic in the application configuration.

User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from stalling the device indefinitely. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Stratix V device. This feature is automatically disabled in the factory configuration image and enabled in the application configuration image. Functional errors must not exist in the factory configuration because they are stored and validated during production and must never be updated remotely.

The user watchdog timer feature is automatically enabled in the application configuration image. If you do not wish to use this feature, disable it during the factory configuration image operation before triggering the reconfiguration to the application configuration image.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29 bits wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The granularity of the timer setting is 2^{17} cycles. The cycle time is based on the frequency of the 12.5-MHz internal oscillator. Table 9–18 lists the operating range of the 12.5-MHz internal oscillator.

Table 9–18. 12.5-MHz Internal Oscillator Specifications *(1)*

Minimum	Typical	Maximum	Units
5.3	7.9	12.5	MHz

Note to Table 9–18:

(1) These values are preliminary.

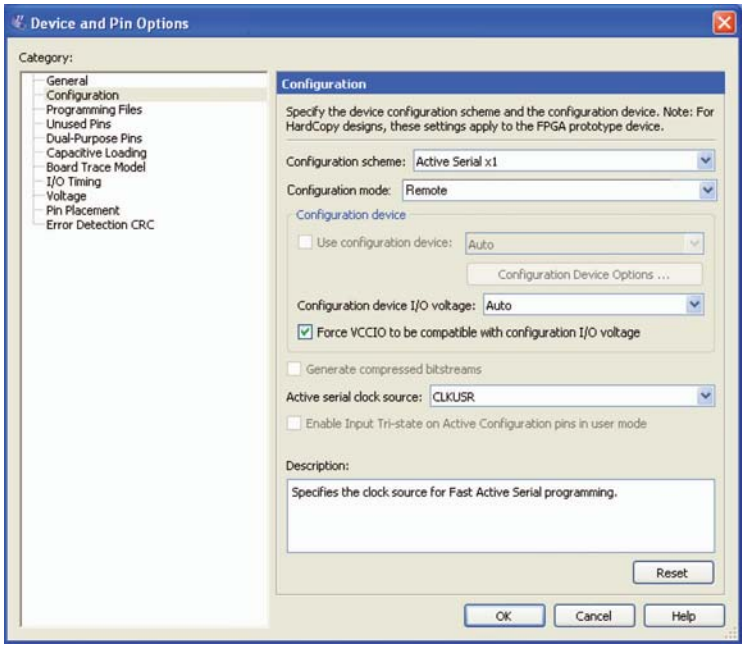
The user watchdog timer begins counting after the application configuration enters device user mode. This timer must be periodically reset by the application configuration before the timer expires by asserting `RU_nRSTIMER`. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. This causes the device to reload the factory configuration image and update the status register to reflect the watchdog timer time-out error.

Enabling the Remote System Update Feature

You can enable remote update for Stratix V devices in the Quartus II software before design compilation (in the Compiler Settings menu). In remote update mode, the **auto-restart configuration after error** option is always enabled. To enable remote update in the project's compiler settings in the Quartus II software, follow these steps:

1. On the Assignments menu, click **Device**. The **Settings** dialog box appears.
2. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.
3. Click the **Configuration** panel.
4. From the Configuration scheme list, select **Active Serial x1** (you can also use **Configuration Device**) (refer to Figure 9–32).
5. From the Configuration mode list, select **Remote** (refer to Figure 9–32).
6. Click **OK**.
7. In the **Settings** dialog box, click **OK**.

Figure 9–32. Enabling Remote Update for Stratix V Devices in the Compiler Settings Menu



ALTREMOTE_UPDATE Megafunction

The ALTREMOTE_UPDATE megafunction provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read and write protocol in the Stratix V device logic. This implementation is suitable for designs that implement the factory configuration functions using a Nios II processor or user logic in the device. Using the megafunction block instead of creating your own logic saves design time and offers more efficient logic synthesis and device implementation.

For more information about the ALTREMOTE_UPDATE megafunction, refer to the *Remote Update Circuitry (ALTREMOTE_UPDATE) Megafunction User Guide*.

Design Security

This section provides an overview of the design security features and their implementation in Stratix V devices using the advanced encryption standard (AES). It also describes the security modes available in Stratix V devices that allow you to use these new features in your designs.

As Stratix V devices continue to play roles in larger and more critical designs in competitive commercial and military environments, it is increasingly important to protect your designs from copying, reverse engineering, and tampering. Stratix V design security supports the following features:

- Enhanced built-in AES decryption block to support 256-key industry-standard design security algorithm (FIPS-197 Certified)
- Volatile and non-volatile key programming support
- Secure operation mode for both volatile and non-volatile key through tamper protection bit setting

- JTAG secure mode is enable through tamper-protection bit
- Supports board level testing
- Supports in-socket key programming for non-volatile key
- Available in all configuration schemes except JTAG
- Supports both remote system upgrades and decompression feature

You can use the design security feature with or without the remote system upgrades or decompression features.

The Stratix V design security feature provides the following security protection for your designs:

- Security against copying—the security key is securely stored in the Stratix V device and cannot be read out through any interface. In addition, as configuration file read-back is not supported in Stratix V devices, your design information cannot be copied.
- Security against reverse engineering—reverse engineering from an encrypted configuration file is very difficult and time consuming because the Stratix V configuration file formats are proprietary and the file contains millions of bits that require specific decryption. In addition, the Stratix V devices are manufactured on the most advanced 28-nm process technology, making this process very difficult.
- Security against tampering—this disables tamper attempts through the JTAG interface. You can enhance this security feature with the tamper protection bit setting. After the tamper protection bit is set, the Stratix V device can only accept configuration files encrypted with the same key. Additionally, programming through the JTAG interface is blocked. This prevents any attempts to tamper with the device from both the JTAG interface and the configuration interface.

When you use compression with the design security feature, the configuration file is first compressed and then encrypted using the Quartus II software. During configuration, the Stratix V device first decrypts and then decompresses the configuration file.

When you use design security with Stratix V devices in a FPP configuration scheme, it requires a different DCLK-to-DATA[] ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

JTAG Secure Mode

When you enable tamper-protection bit, Stratix V devices are in JTAG secure mode after power-up. During JTAG secure mode, many JTAG instructions are disabled. Stratix V devices only allow mandatory JTAG 1149.1 and 1149.6 instructions to be exercised. These instructions are SAMPLE/PRELOAD, BYPASS, EXTEST, and optional instructions such as IDCODE and SHIFT_EDERROR_REG.

To enable the access of other JTAG instructions such as USERCODE, HIGHZ, CLAMP, PULSE_NCONFIG, and CONFIG_IO, you must issue UNLOCK instruction to deactivate the JTAG secure mode. You can issue LOCK instruction to put the device back into JTAG secure mode. Both the LOCK and UNLOCK instructions can only be issued through JTAG core access during user mode.

For more information about JTAG binary instruction code related to the LOCK and UNLOCK instructions, refer to the *JTAG Boundary-Scan Testing in Stratix V Devices* chapter.

Security Key Types

Stratix V devices offer two types of keys—volatile and non-volatile. Table 9–19 lists the differences between the volatile key and non-volatile key.

Table 9–19. Security Key Types

Key Types	Key Programmability	Power Supply for Key Storage	Programming Method
Volatile Key	■ Re-programmable ■ Erasable	Required external battery, V_{CCBAT} (1)	On-board
Non-volatile Key	One-time programming	Does not require an external battery	On-board and in-socket programming (2)

Notes to Table 9–19:

(1) V_{CCBAT} is a dedicated power supply for volatile key storage and not shared with other on-chip power supplies, such as V_{CCIO} or V_{CCPGM}. V_{CCBAT} continuously supplies power to the volatile register regardless of the on-chip supply condition.

(2) In-socket programming is offered through third party vendors.

Both non-volatile and volatile key programming offers protection from reverse engineering and copying. If you set the tamper-protection bit, the design is also protected from tampering.

Perform key programming through the JTAG interface. Also, ensure that the nSTATUS pin is released high before any key-programming attempts.

To clear the volatile key, issue the KEY_CLR_VREG JTAG instruction. To verify the volatile key has been cleared, issue the KEY_VERIFY JTAG instruction.

For more information about KEY_CLR_VREG and KEY_VERIFY JTAG instructions, refer to the *JTAG Boundary-Scan Testing in Stratix V Devices* chapter.

For more information about battery specifications, refer to the *Stratix V Device Datasheet*.

For more information about the V_{CCBAT} pin connection recommendations, refer to the *Stratix V Device Family Pin Connection Guidelines*.

Security Modes

Table 9–20 lists the security modes available in Stratix V devices.

Table 9–20. Supported Security Modes

Security Mode	Tamper protection Bit Setting	Device Accepts Unencrypted File	Device Accepts Encrypted File	Security Level
No-key	—	Yes	No	—
Volatile Key	—	Yes (2)	Yes	Secure
Volatile Key with Tamper Protection Bit Set	Set (1)	No	Yes	Secure with tamper resistant
Non-volatile Key	—	Yes (2)	Yes	Secure
Non-volatile Key with Tamper Protection Bit Set	Set (1)	No	Yes	Secure with tamper resistant

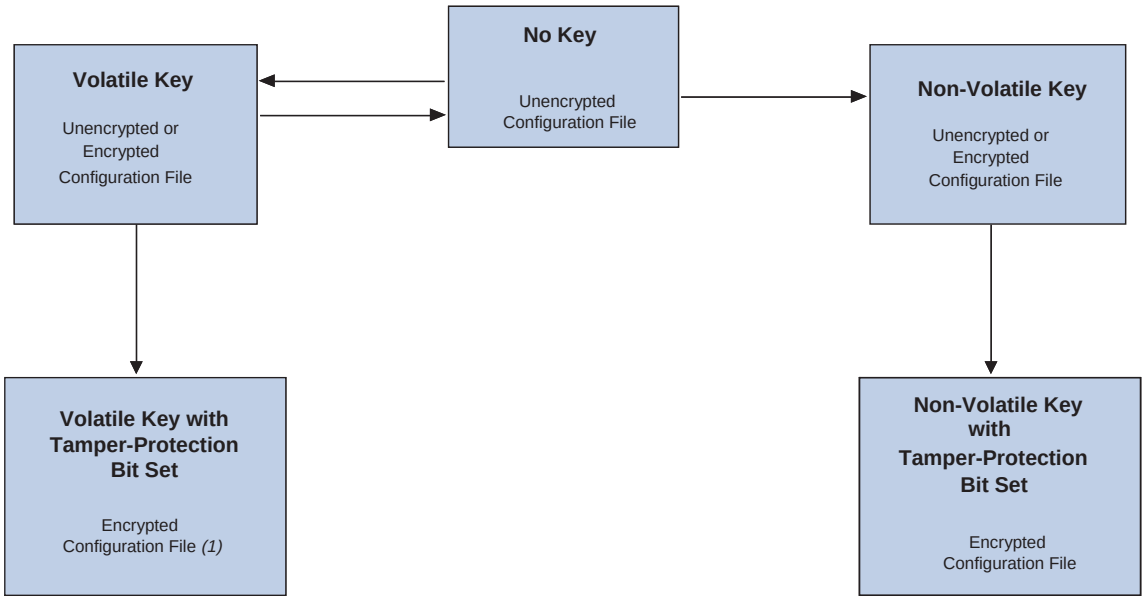
Notes to Table 9–20:

(1) Enabling the tamper protection bit disables test mode in Stratix V devices and disables programming through the JTAG interface. This process is irreversible and prevents Altera from carry-out failure analysis. Contact Altera Technical Support to enable the tamper protection bit.

(2) Use the unencrypted configuration bitstream support only for board-level testing.

Figure 9–33 shows the sequence of the security modes available in Stratix V devices.

Figure 9–33. Stratix V Security Modes—Sequence and Restrictions



Note to Figure 9–33:

(1) Stratix V devices do not accept the encrypted configuration file if the volatile key is erased. You must use the volatile key without tamper-protection bit set to reprogram the key if the volatile key in Stratix V device is erased.

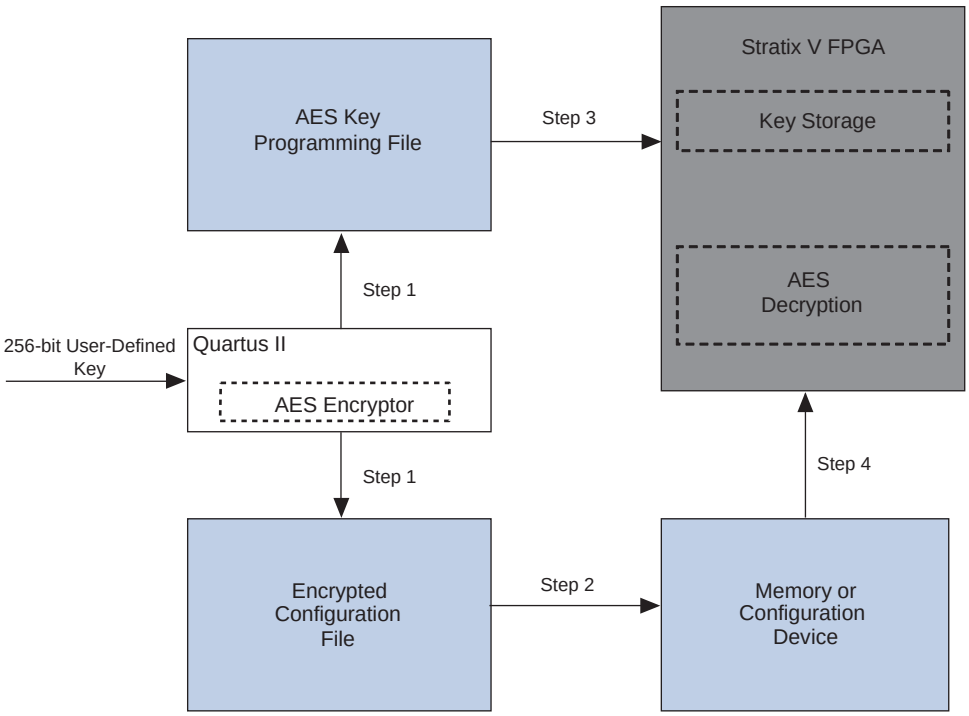
Design Security Implementation Steps

Stratix V devices are SRAM-based devices. To provide design security, Stratix V devices require a 256-bit security key for configuration bitstream design security. To carry out secure configuration, follow these steps:

1. The Quartus II software generates the design security key programming file and encrypts the configuration data using the user-defined 256-bit key.
2. Store the encrypted configuration file in the external memory.
3. Program the AES key programming file into the Stratix V device through a JTAG interface.
4. Configure the Stratix V device. At system power-up, the external memory device sends the encrypted configuration file to the Stratix V device.

Figure 9–34 shows the design security implementation steps.

Figure 9–34. Design Security Implementation Steps



Document Revision History

Table 9–21 lists the revision history for this chapter.

Table 9–21. Document Revision History

Date	Version	Changes
June 2012	1.7	■ Added MAX V devices. ■ Updated Figure 9–2, Figure 9–3, Figure 9–11, Figure 9–16, Figure 9–17, Figure 9–20, and Figure 9–23. ■ Updated Table 9–4, Table 9–5, Table 9–7, Table 9–11, and Table 9–12. ■ Updated "MSEL Pin Settings" and "FPP Multi-Device Configuration" sections.
February 2012	1.6	■ Updated "Security Key Types" section. ■ Updated Table 9–10.
December 2011	1.5	■ Updated "FPP Configuration Timing", "JTAG Secure Mode", and "Security Key Types" sections. ■ Updated Table 9–8.
November 2011	1.4	■ Updated Table 9–5, Table 9–9, and Table 9–14. ■ Updated Figure 9–8, Figure 9–9, and Figure 9–21. ■ Updated "AS Multi-Device Configuration" and "Active Serial Configuration (Serial Configuration Devices)" sections.
May 2011	1.3	■ Chapter moved to volume 2 for the 11.0 release. ■ Added "Remote System Upgrade Using EPCQ 256" and "JTAG Secure Mode" sections. ■ Updated Table 9–5. ■ Updated "Configuration", "Configuration Error", "Programming EPCS and EPCQ", "JTAG Configuration", "Remote Update Mode", and "Design Security" sections. ■ Minor text edits.
January 2011	1.2	■ Updated Table 9–7, Table 9–8, Table 9–12, and Table 9–14. ■ Updated Figure 9–15 and Figure 9–21. ■ Updated "User Watchdog Timer", "DCLK-to-DATA[] Ratio for FPP configuration", "V_{CCPD} Pin", "POR Delay Specification", and "Programming EPCS and EPCQ" sections.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

10. SEU Mitigation in Stratix V Devices

SV51011-1.5

This chapter describes how to activate and use the error detection cyclic redundancy check (CRC) feature when your Stratix® V device is in user mode and how to recover from configuration errors caused by CRC errors. The error detection feature is enhanced in the Stratix V devices.

This chapter contains the following sections:

- "Error Detection Fundamentals" on page 10–1
- "Configuration Error Detection" on page 10–2
- "User Mode Error Detection and Correction" on page 10–2
- "Error Detection Pin Description" on page 10–5
- "Error Detection Block" on page 10–6
- "Error Detection Timing" on page 10–8
- "Software Support" on page 10–11
- "Recovering From CRC Errors" on page 10–11

In critical applications such as avionics, telecommunications, system control, and military applications, it is important to be able to do the following:

- Confirm that the configuration data stored in a Stratix V device is correct.
- Alert the system to the occurrence of a configuration error.

For Stratix V devices, the error detection CRC feature is provided in the Quartus® II software starting with version 10.0.

Using the error detection CRC feature for the Stratix V device family has no impact on fitting or performance.

Error Detection Fundamentals

Error detection determines if the data received through a medium is corrupted during transmission. To accomplish this, the transmitter uses a function to calculate a checksum value for the data and appends the checksum to the original data frame. The receiver uses the function to calculate a checksum for the received data frame and compares the received checksum to the transmitted checksum. If the two checksum values are equal, the received data frame is correct and no data corruption occurred during transmission or storage.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

The error detection CRC feature uses the same concept. When Stratix V devices are successfully configured and in user mode, the error detection CRC feature ensures the integrity of the configuration data.

Configuration Error Detection

In configuration mode, a frame-based 16-bit configuration CRC is stored in the configuration data and contains the CRC value for each data frame.

During configuration, the Stratix V device calculates the 16-bit configuration CRC value based on the frame of data that is received and compares it against the pre-calculated 16-bit configuration CRC value in the data stream. If the 16-bit configuration CRC values do not match, nSTATUS is set low. Configuration continues until either the device detects an error or configuration is complete.

User Mode Error Detection and Correction

Stratix V devices offer on-chip circuitry for automated single event upset (SEU) detection. Some applications require the device to operate error-free in high-neutron flux environments require periodic checks to ensure continued data integrity. The error detection CRC feature ensures data reliability and is one of the best options for mitigating SEU.

You can implement the error detection CRC feature with existing circuitry in Stratix V devices, eliminating the need for external logic. Stratix V devices have built-in error detection circuitry to detect data corruption by soft errors in the configuration random access memory (CRAM) cells. This feature allows all CRAM contents to be read and verified to match a configuration-computed 32-bit error detection CRC value. Soft errors are changes in a CRAM's bit state due to an ionizing particle.

To enable the error detection process when the device transitions into user mode, turn on the **Enable Error Detection CRC_ERROR pin** option on the **Error Detection CRC** page of the **Device and Pin Options** dialog box in the Quartus II software.

The error detection capability continuously calculates the 32-bit error detection CRC value of the configured CRAM bits and compares it with the configuration-computed 32-bit error detection CRC value. The 32-bit error detection CRC value is computed during the configuration stage. The error detection circuitry generates 32 CRC check bits per frame and then stores them in the CRAM. If the 32-bit error detection CRC values match, there is no error in the current configuration CRAM bits. The process of error detection continues until the device is reset by setting nCONFIG low.

A single 32-bit error detection CRC calculation is done on a per frame basis. After the error detection circuitry has finished the CRC calculation for a frame, the resulting 32-bit signature is 0×00000000. If the error detection circuitry detects no CRAM bit errors in a frame, the output signal CRC_ERROR is set to low. If the circuitry detects a CRAM bit error in a frame in the device, the resulting signature is non-zero and the error detection circuitry starts searching for the error bit location.

The error detection circuitry in Stratix V devices calculates CRC check bits for each frame and pulls the CRC_ERROR pin high when it detects bit errors in the chip. Within a frame, it can detect all single-bit, double-bit, triple-bit, quadruple-bit, and quintuple-bit errors. The probability of more than five CRAM bits being flipped by a SEU is very low. In general, the probability of detection for all error patterns is 99.9999%.

The error detection circuitry reports the bit location and determines the type of error for single-bit errors or double-adjacent errors. The probability of other error patterns is very low and the reporting of bit location is not guaranteed.

You can also read the error bit location through the JTAG and the core interface. Before the error detection circuitry detects the next error in another frame, you must shift erroneous bits out from the error message register (EMR) with either the JTAG instruction, SHIFT_EDERROR_REG, or the core interface. The CRC circuitry continues to run, and if an error is detected, you must decide whether to complete the reconfiguration or to ignore the CRC error.

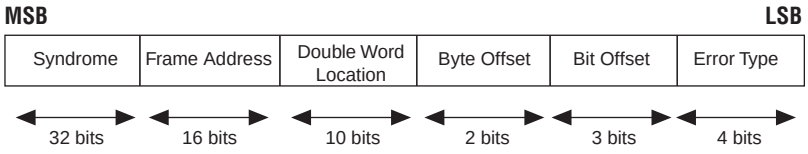
Table 10–1 lists the instruction code for the SHIFT_EDERROR_REG JTAG instruction.

Table 10–1. SHIFT_EDERROR_REG JTAG Instruction

JTAG Instruction	Instruction Code	Description
SHIFT_EDERROR_REG	00 0001 0111	The JTAG instruction connects the EMR to the JTAG pin in the error detection block between the TDI and TDO pins.

Figure 10–1 shows the content of the EMR.

Figure 10–1. Error Message Register



The type of error is identified in the first four bits of the EMR. Table 10–2 lists the error types represented in the EMR.

Table 10–2. Error Type in Error Message Register

Error Type				Description
Bit 3	Bit 2	Bit 1	Bit 0	
0	0	0	0	No CRC error.
0	0	0	1	Location of a single-bit error is identified.
0	0	1	0	Location of a double-adjacent error is identified.
1	1	1	1	There is more than one error.
Others				Reserved.

For more information about the timing requirement to shift out error information from the EMR, refer to "Error Detection Timing" on page 10–8.

The error detection circuitry continues to calculate the 32-bit error detection CRC value and 32-bit signatures for the next frame of data regardless of whether an error has occurred in the current frame or not. You must monitor the CRC_ERROR signal and take the appropriate actions if a CRC error occurs.

The error detection circuitry in Stratix V devices uses a 32-bit CRC-ANSI standard (32-bit polynomial) as the CRC generator. The computed 32-bit CRC signature for each frame is stored in the CRAM. The total storage size is 32 (number of bits per frame) × the number of frames.

The Stratix V device error detection CRC feature does not check memory blocks and I/O buffers. Thus, the CRC_ERROR signal may stay solid high or low, depending on the error status of the previously checked CRAM frame. The I/O buffers are not verified during error detection because these bits use flipflops as storage elements that are more resistant to soft errors when compared with CRAM cells. MLAB and M20K memory blocks support parity bits that are used to check the contents of memory blocks for any error.

For more information about error detection in Stratix V memory blocks, refer to the *Memory Blocks in Stratix V Devices* chapter.

In Stratix V, in addition to the error detection capability, the error detection circuitry also supports error correction or internal scrubbing, which is an ability to internally correct soft errors that have been detected. This is done on a per-frame basis. If internal scrubbing is enabled, the device corrects single-bit error or double-adjacent error in the CRAM bits while the frame where the error occurred is still operational and the device is still running.

To provide testing capability of the error detection block, a JTAG instruction, EDERROR_INJECT, is provided. This instruction is able to change the content of the 47-bit JTAG fault injection register used for error injection in Stratix V devices, thereby enabling testing of the error detection block.

You can only execute the EDERROR_INJECT JTAG instruction when the device is in user mode.

Table 10–3 lists the EDERROR_INJECT JTAG instruction.

Table 10–3. EDERROR_INJECT JTAG Instruction

JTAG Instruction	Instruction Code	Description
EDERROR_INJECT	00 0001 0101	This instruction controls the 47-bit JTAG fault injection register used for error injection.

You can create a Jam™ file (**.jam**) to automate the testing and verification process. This allows you to verify the CRC functionality in-system and on-the-fly, without having to reconfigure the device. You can then switch to the CRC circuit to check for real errors induced by a SEU.

You can introduce a single error or double errors adjacent to each other to the configuration memory. This provides an extra way to facilitate design verification and system fault tolerance characterization. Use the JTAG fault injection register with the EDERROR_INJECT JTAG instruction to flip the readback bits. The Stratix V device is then forced into error test mode. Altera recommends reconfiguring the device after the test completes.

You can only introduce error injection in the first data frame, but you can monitor the error information at any time. For more information about the JTAG fault injection register and fault injection register, refer to "Error Detection Registers" on page 10–7.

Table 10–4 lists how the fault injection register is implemented and describes error injection.

Table 10–4. Fault Injection Register

Description	Bit[46..43]					Bit[42..32]	Bit[31..0]
	Error Type				Error Injection Type	Byte Location of the Injected Error	Error Byte Value
	Bit[46]	Bit[45]	Bit[44]	Bit[43]			
Content	0	0	0	0	No error injection	Depicts the location of the injected error in the first data frame.	Depicts the location of the bit error and corresponds to the error injection type selection.
	0	0	0	1	Single error injection		
	0	0	1	0	Double-adjacent error injection		
	Others				Reserved		

Error Detection Pin Description

Table 10–5 lists the CRC_ERROR pin.

Table 10–5. CRC_ERROR Pin Description

Pin Name	Pin Type	Description
CRC_ERROR	I/O, output, or output open-drain	Active high signal indicating that the error detection circuit has detected errors in the configuration CRAM bits. This is an optional pin and is used when you enable the error detection CRC circuit. When you disable the error detection CRC circuit, it is an user I/O pin. When using the WYSIWYG function, you can route the crcerror port from the WYSIWYG atom to the dedicated CRC_ERROR pin or any user I/O. To route the crcerror port to user I/O, you must insert a D-type flipflop in between the crcerror port and the I/O. By default, the Quartus II software sets the CRC_ERROR pin as output open-drain when you enable the error detection CRC circuitry. By option, you can set this pin to be an output by turning off the **Enable open-drain on CRC_ERROR pin** option on the **Error Detection CRC** page of the **Device and Pin Option** dialog box in the Quartus II software. If the CRC_ERROR pin is used as an output, you must ensure that the V_{CCIO} of the bank in which the pin resides meet the input voltage specification of the system receiving the signal. Using the pin as an open-drain provides an advantage on the voltage leveling. To use the CRC_ERROR pin as an open-drain, you can tie this pin to V_{CCPGM} through a 10-kΩ resistor. Alternatively, depending on the input voltage specification of the system receiving the signal, tie this pin to a different pull-up voltage.

Error Detection Block

The error detection block contains the logic necessary to calculate the 32-bit error detection CRC signature for the configuration CRAM bits in the Stratix V device.

The CRC circuit continues running even if an error occurs. When a CRC error occurs, the device sets the CRC_ERROR pin high. Table 10–6 lists the two types of CRC detection that check the configuration bits.

Table 10–6. Two Types of CRC Detection

User Mode CRC Error Detection	Configuration CRC Error Detection
■ This is the CRAM error checking ability (32-bit error detection CRC) during user mode for use by the CRC_ERROR pin. ■ For each frame of data, the pre-calculated 32-bit error detection CRC enters the CRC circuit at the end of the frame data and determines whether there is an error or not. ■ If an error occurs, the search engine finds the location of the error. ■ The error messages can be shifted out through the JTAG instruction or core interface logics while the error detection block continues running. ■ The JTAG interface reads out the 32-bit error detection CRC result for the first frame and also shifts the 32-bit error detection CRC bits to the 32-bit error detection CRC storage registers for test purposes. ■ You can deliberately introduce single error or double-adjacent error to the configuration memory for testing and design verification.	■ This is the 16-bit configuration CRC that is embedded in every configuration data frame. ■ During configuration, after a frame of data is loaded into the Stratix V device, the pre-computed configuration CRC is shifted into the CRC circuitry. ■ At the same time, the 16-bit configuration CRC value for the data frame shifted-in is calculated. If the pre-computed configuration CRC and calculated configuration CRC values do not match, nSTATUS is set low. Every data frame has a 16-bit configuration CRC; therefore, there are many 16-bit configuration CRC values for the whole configuration bitstream as there are many data frames. Every device has different lengths of the configuration data frame.

Error Detection Registers

This section focuses on the user mode CRC error detection. There is one set of 32-bit registers in the error detection circuitry that stores the computed CRC signature. A non-zero value on the syndrome register causes the CRC_ERROR pin to be set high.

Figure 10–2 shows the error detection circuitry, syndrome registers, and error injection block.

Figure 10–2. Error Detection Circuitry, Syndrome Registers, and Error Injection Block

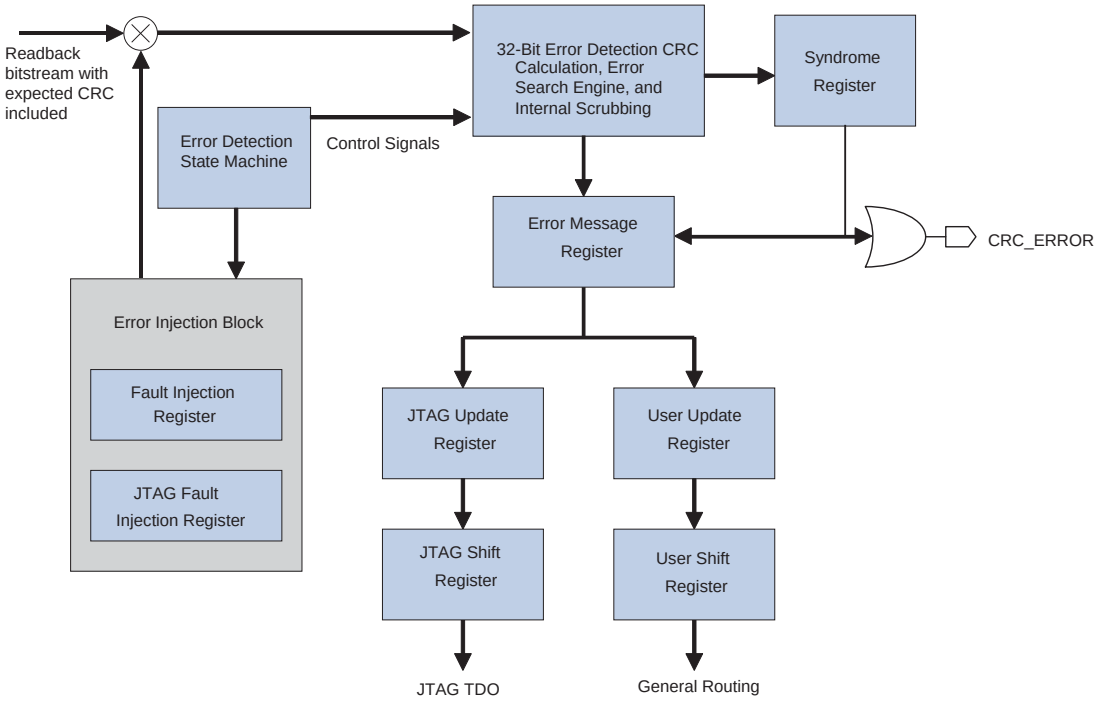


Table 10–7 lists the registers shown in Figure 10–2.

Table 10–7. Error Detection Registers (Part 1 of 2)

Register	Description
Syndrome Register	This 32-bit register contains the CRC signature of the current frame through the error detection verification cycle. The CRC_ERROR signal is derived from the contents of this register.
Error Message Register	This 67-bit register contains information on the error type, location of the error, and the actual syndrome. The types of errors and location reported are single- and double-adjacent bit errors. The location bits for other types of errors are not identified by the EMR. The content of the register is shifted out through the SHIFT_EDERROR_REG JTAG instruction or to the core through the core interface.
JTAG Update Register	This 67-bit register is automatically updated with the contents of the EMR one cycle after this register content is validated. It includes a clock enable, which must be asserted prior to being sampled into the JTAG shift register. This requirement ensures that the JTAG Update Register is not being written into by the contents of the EMR at the same time that the JTAG shift register is reading its contents.

Table 10–7. Error Detection Registers (Part 2 of 2)

Register	Description
User Update Register	This 67-bit register is automatically updated with the contents of the EMR one cycle after this register content is validated. It includes a clock enable, which must be asserted prior to being sampled into the user shift register. This requirement ensures that the user update register is not being written into by the contents of the EMR at exactly the same time that the user shift register is reading its contents.
JTAG Shift Register	This 67-bit register is accessible by the JTAG interface and allows the contents of the JTAG update register to be sampled and read out by SHIFT_EDERROR_REG JTAG instruction.
User Shift Register	This 67-bit register is accessible by the core logic and allows the contents of the user update register to be sampled and read by user logic.
JTAG Fault Injection Register	This 47-bit register is fully controlled by the EDERROR_INJECT JTAG instruction. This register holds the information of the error injection that you want in the bitstream.
Fault Injection Register	The content of the JTAG fault injection register is loaded into this 47-bit register when it is updated.

Error Detection Timing

When you enable the error detection CRC feature through the Quartus II software, the device automatically activates the CRC error detection process after entering user mode.

If an error is detected within a frame, CRC_ERROR is driven high at the end of the error location search, after the EMR is updated. At the end of this cycle, the CRC_ERROR pin is pulled low for a minimum of 32 clock cycles. If the next frame contains an error, CRC_ERROR is driven high again after the EMR is overwritten by the new value. You can start to unload the error message on each rising edge of the CRC_ERROR pin. Error detection runs until the device is reset.

The error detection circuitry runs off an internal configuration oscillator with a divisor that sets the maximum frequency. Table 10–8 lists the minimum and maximum error detection frequencies.

Table 10–8. Minimum and Maximum Error Detection Frequencies

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (n)
Stratix V	100 MHz/2^n	50 MHz	390 KHz	1, 2, 3, 4, 5, 6, 7, 8

You can set a lower clock frequency by specifying a division factor in the Quartus II software (refer to "Software Support" on page 10–11). The divisor is a power of two (2), where *n* is between 1 and 8. The divisor ranges from 2 through 256 (Equation 10–1).

Equation 10–1.

$$\text{Error Detection Frequency} = \frac{100\text{MHz}}{2^n}$$

The error detection frequency reflects the frequency of the error detection process for a frame because the CRC calculation in Stratix V devices is done on a per-frame basis.

The EMR is updated whenever an error occurs. If the error location and message are not shifted out before the next error location is found, the previous error location and message are overwritten by the new information. To avoid this, you must shift these bits out within one frame of the CRC verification. The minimum interval time between each update for the EMR depends on the device and the error detection clock frequency. However, slowing down the error detection clock frequency slows down the error recovery time for the SEU event.

Table 10–9 lists the estimated minimum interval time between each update for the EMR for Stratix V devices.

Table 10–9. Minimum Update Interval for Error Message Register—Preliminary

Family	Device	Package	Timing Interval (µs)
Stratix V GX	5SGXA3	EH29-H780	2.43
		HF35-F1152	2.43
		KF35-F1152	2.99
		KF40-F1517/KH40-H1517	2.99
	5SGXA4	(1)	2.99
	5SGXA5	(1)	3.59
	5SGXA7	(1)	3.59
	5SGXA9	(1)	4.87
	5SGXAB	(1)	4.87
	5SGXB5	(1)	3.73
	5SGXB6	(1)	3.73
Stratix V GT	5SGTC5	(1)	3.59
	5SGTC7	(1)	3.59
Stratix V GS	5SGSD3	(1)	2.43
	5SGSD4	EH29-H780	2.43
		HF35-F1152	2.43
		KF40-F1517/KH40-H1517	2.99
	5SGSD5	(1)	2.99
	5SGSD6	(1)	4.21
	5SGSD8	(1)	4.21
Stratix V E	5SEE9	(1)	4.87
	5SEEB	(1)	4.87

Note to Table 10–9:

(1) Applies to all packages of this device.

The CRC calculation time for the error detection circuitry to check from the first until the last frame depends on the device and the error detection clock frequency.

The minimum CRC calculation time is calculated using the maximum error detection frequency with a divisor factor of 1. The maximum CRC calculation time is calculated using the minimum error detection frequency with a divisor factor of 8. You can estimate the CRC calculation time for other valid divisors (n) for a specific device density using the Equation 10–2:

Equation 10–2.

$$\text{CRC Calculation Time} = \frac{\text{Minimum calculation time}}{2} \times 2^n$$

Table 10–10 lists the minimum and maximum estimated clock frequency time for each CRC calculation for Stratix V devices.

Table 10–10. CRC Calculation Time—Preliminary

Family	Device	Package	Minimum Time (ms)	Maximum Time (s)
Stratix V GX	5SGXA3	EH29-H780	43	22.288
		HF35-F1152	43	22.288
		KF35-F1152	67	34.809
		KF40-F1517/KH40-H1517	67	34.809
	5SGXA4	(1)	67	34.809
	5SGXA5	(1)	84	43.628
	5SGXA7	(1)	84	43.628
	5SGXA9	(1)	105	54.832
	5SGXAB	(1)	105	54.832
	5SGXB5	(1)	84	43.628
	5SGXB6	(1)	84	43.628
Stratix V GT	5SGTC5	(1)	84	43.628
	5SGTC7	(1)	84	43.628
Stratix V GS	5SGSD3	(1)	43	22.288
	5SGSD4	EH29-H780	43	22.288
		HF35-F1152	43	22.288
		KF40-F1517/KH40-H1517	67	34.809
	5SGSD5	(1)	67	34.809
	5SGSD6	(1)	91	47.128
	5SGSD8	(1)	91	47.128
Stratix V E	5SEE9	(1)	105	54.832
	5SEEB	(1)	105	54.832

Note to Table 10–10:

(1) Applies to all packages of this device.

Software Support

The Quartus II software, starting with version 10.0, supports the error detection CRC feature for Stratix V devices. Enabling this feature in the **Device and Pin Options** dialog box generates the CRC_ERROR output to the optional dual-purpose CRC_ERROR pin.

To enable the error detection feature using CRC, perform the following steps:

1. Open the Quartus II software and load a project using a Stratix V device.
2. On the Assignments menu, click **Device**. The **Device** dialog box appears.
3. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.
4. In the **Category** list, click **Error Detection CRC**.
5. Turn on **Enable Error Detection CRC_ERROR pin**.
6. By default, the Quartus II software sets the CRC_ERROR pin as output open-drain when you enable the error detection CRC circuitry. You can set this pin to be an output by turning off the **Enable open-drain on CRC_ERROR pin**.
7. In the **Divide error check frequency by** box, enter a valid divisor as shown in Table 10–8 on page 10–8.
8. To enable or disable the on-chip error correction feature, turn on or turn off **Enable internal scrubbing**.
9. Click **OK**.

Recovering From CRC Errors

The system that the Stratix V device resides in must control device reconfiguration. After detecting an error on the CRC_ERROR pin, strobing the nCONFIG signal low directs the system to perform the reconfiguration at a time when it is safe.

When the data bit is rewritten with the correct value by reconfiguring the device, the device functions correctly.

While soft errors are uncommon in Altera® devices, certain high-reliability applications may require a design to account for these errors.

Document Revision History

Table 10–11 lists the revision history for this chapter.

Table 10–11. Document Revision History

Date	Version	Changes
June 2012	1.5	Minor text edits.
February 2012	1.4	Updated Table 10–9 and Table 10–10.
November 2011	1.3	■ Updated Table 10–9 and Table 10–10. ■ Updated "User Mode Error Detection and Correction" section.
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 10–9 and Table 10–10. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

11. JTAG Boundary-Scan Testing in Stratix V Devices

SV51012-1.5

This chapter describes the boundary-scan test (BST) features that are supported in Stratix® V devices.

Stratix V devices support IEEE Std. 1149.1 and IEEE Std. 1149.6. The IEEE Std. 1149.6 is only supported on the high-speed serial interface (HSSI) transceivers in Stratix V devices. IEEE Std. 1149.6 enables board-level connectivity checking between transmitters and receivers that are AC coupled (connected with a capacitor in series between the source and destination).

This chapter includes the following sections:

- "IEEE Std. 1149.6 Boundary-Scan Register" on page 11–2
- "BST Operation Control" on page 11–3
- "I/O Voltage Support in a JTAG Chain" on page 11–5
- "Boundary-Scan Description Language Support" on page 11–6

For more information about the following IEEE Std. 1149.1 BST features, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*:

- IEEE Std. 1149.1 BST architecture and circuitry
- IEEE Std. 1149.1 boundary-scan register
- IEEE Std. 1149.1 BST guidelines
- Test access port (TAP) controller state-machine

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

IEEE Std. 1149.6 Boundary-Scan Register

The boundary-scan cells (BSCs) for HSSI transmitters (GXB_TX[p,n]) and receivers/input clock buffer (GXB_RX[p,n])/(REFCLK[p,n]) in Stratix V devices are different from the BSCs for the I/O pins.

Figure 11–1 shows the Stratix V HSSI transmitter BSC.

Figure 11–1. HSSI Transmitter BSC with IEEE Std. 1149.6 BST Circuitry for Stratix V Devices

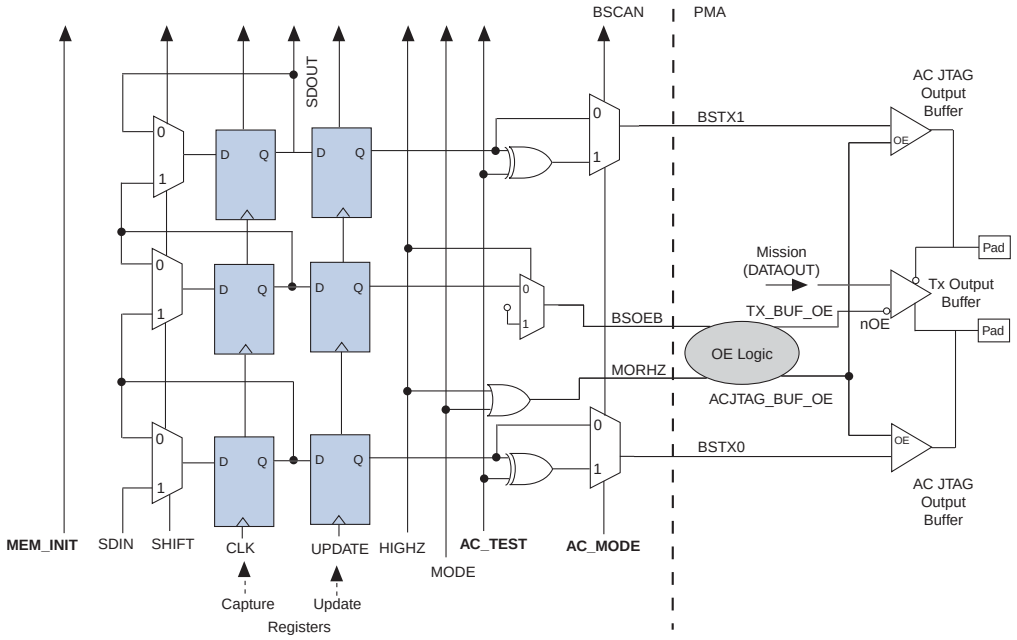
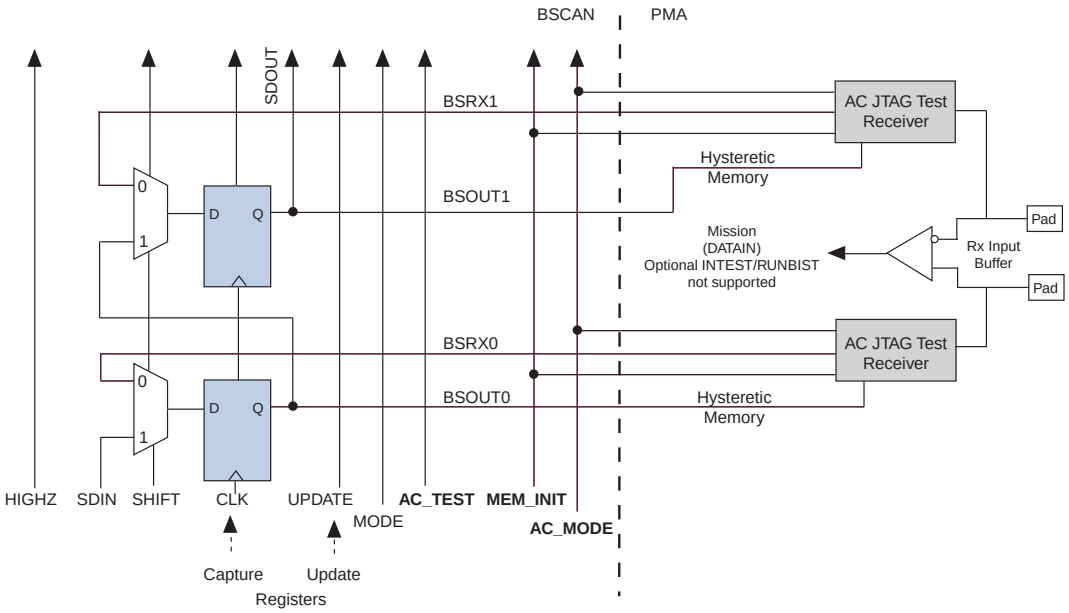


Figure 11–2 shows the Stratix V HSSI receiver/input clock buffer BSC.

Figure 11–2. HSSI Receiver/Input Clock Buffer BSC with IEEE Std. 1149.6 BST Circuitry for Stratix V Devices



BST Operation Control

Table 11–1 lists the IDCODE information for Stratix V devices.

Table 11–1. 32-Bit IDCODE Information for Stratix V Devices—Preliminary

Family	Device	IDCODE (32 Bits) (1)			
		Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) (2)
Stratix V GX	5SGXA3 (3)	0000	0010 1001 0100 0111	000 0110 1110	1
	5SGXA3 (4)	0000	0010 1001 0010 0001	000 0110 1110	1
	5SGXA4	0000	0010 1001 0010 0111	000 0110 1110	1
	5SGXA5	0000	0010 1001 0001 0011	000 0110 1110	1
	5SGXA7	0000	0010 1001 0000 0011	000 0110 1110	1
	5SGXA9	0000	0010 1001 0100 0101	000 0110 1110	1
	5SGXAB	0000	0010 1001 0010 0101	000 0110 1110	1
	5SGXB5	0000	0010 1001 0001 0010	000 0110 1110	1
	5SGXB6	0000	0010 1001 0000 0010	000 0110 1110	1
	5SGXB9	0000	0010 1001 0001 0101	000 0110 1110	1
	5SGXBB	0000	0010 1001 0000 0101	000 0110 1110	1
Stratix V GT	5SGTC5	0000	0010 1001 0010 0011	000 0110 1110	1
	5SGTC7	0000	0010 1001 0100 0011	000 0110 1110	1
Stratix V GS	5SGSD3	0000	0010 1001 0001 0001	000 0110 1110	1
	5SGSD4 (5)	0000	0010 1001 0000 0001	000 0110 1110	1
	5SGSD4 (6)	0000	0010 1001 0001 0111	000 0110 1110	1
	5SGSD5	0000	0010 1001 0000 0111	000 0110 1110	1
	5SGSD6	0000	0010 1001 0001 0100	000 0110 1110	1
	5SGSD8	0000	0010 1001 0000 0100	000 0110 1110	1
Stratix V E	5SEE9	0000	0010 1001 1001 0101	000 0110 1110	1
	5SEEB	0000	0010 1001 1000 0101	000 0110 1110	1

Notes to Table 11–1:

(1) The MSB is on the left.
(2) The LSB of the IDCODE is always 1.
(3) The IDCODE is applicable for KF35 and KF40 packages only.
(4) The IDCODE is applicable for EH29 and HF35 packages only.
(5) The IDCODE is applicable for EH29 and HF35 packages only.
(6) The IDCODE is applicable for the KF40 package only.

Table 11–2 lists the JTAG instructions that are supported by Stratix V devices.

Table 11–2. JTAG Instruction Supported by Stratix V Devices (Part 1 of 2)

JTAG Instruction	Instruction Code	Description
SAMPLE/PRELOAD	00 0000 0101	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation and permits an initial data pattern to be an output at the device pins. Also used by the SignalTap™ II Embedded Logic Analyzer.
EXTEST (1)	00 0000 1111	Allows the external circuit and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	11 1111 1111	Places the 1-bit bypass register between the TDI and TDO pins, allowing the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.
USERCODE	00 0000 0111	Selects the 32-bit USERCODE register and places it between the TDI and TDO pins, allowing USERCODE to be serially shifted out of TDO.
IDCODE	00 0000 0110	Selects the IDCODE register and places it between the TDI and TDO pins, allowing IDCODE to be serially shifted out of TDO. IDCODE is the default instruction at power up and in the TAP RESET state.
HIGHZ (1)	00 0000 1011	Places the 1-bit bypass register between the TDI and TDO pins, allowing the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while tri-stating all of the I/O pins.
CLAMP (1)	00 0000 1010	Places the 1-bit bypass register between the TDI and TDO pins, allowing the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while holding I/O pins to a state defined by the data in the boundary-scan register.
PULSE_NCONFIG	00 0000 0001	Emulates pulsing the nCONFIG pin low to trigger reconfiguration even though the physical pin is unaffected.
CONFIG_IO	00 0000 1101	Allows I/O reconfiguration through JTAG ports using IOCSR for JTAG testing. This is executed after or during configurations. The nSTATUS pin must go high before you can issue the CONFIG_IO instruction.
LOCK	01 1111 0000	Disables the access by JTAG instructions and places the device into JTAG secure mode.
UNLOCK	11 0011 0001	Enables access for JTAG instructions that are not disabled by security fuses.
KEY_CLR_VREG	00 0010 1001	Clears the non-volatile key.
KEY_VERIFY	00 0001 0011	Verifies the non-volatile key has been cleared.
EXTEST_PULSE	00 1000 1111	Enables board-level connectivity checking between the transmitters and receivers that are AC coupled by generating three output transitions: ■ Driver drives data on the falling edge of TCK in the UPDATE_IR/DR state. ■ Driver drives inverted data on the falling edge of TCK after entering the RUN_TEST/IDLE state. ■ Driver drives data on the falling edge of TCK after leaving the RUN_TEST/IDLE state.

Table 11–2. JTAG Instruction Supported by Stratix V Devices (Part 2 of 2)

JTAG Instruction	Instruction Code	Description
EXTEST_TRAIN	00 0100 1111	Behaves the same as the EXTEST_PULSE instruction except that the output continues to toggle on the TCK falling edge as long as the TAP controller is in the RUN_TEST/IDLE state.

Note to Table 11–2:

(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

If any of the security fuses is blown, some of the JTAG instructions are disabled. UNLOCK instruction enables the JTAG instructions that are not blocked by these fuses. LOCK instruction disables these JTAG instructions, returning to the "JTAG secure" state. You can only issue the LOCK or UNLOCK instruction from the JTAG core access.

The IDCODE instruction is the default instruction when the TAP controller is in the reset state. Without loading any instructions, you can go to the SHIFT_DR state and shift out the JTAG device ID.

If the device is in reset state, when the nCONFIG or nSTATUS signal is low, the device IDCODE might not be read correctly. To read the device IDCODE correctly, you must issue the IDCODE JTAG instruction only when the nCONFIG and nSTATUS signals are high.

IEEE Std. 1149.6 mandates the addition of two new instructions—EXTEST_PULSE and EXTEST_TRAIN. These two instructions enable edge-detecting behavior on the signal path containing the HSSI pins. These instructions implement new test behaviors for the HSSI pins and simultaneously behave identically to the IEEE Std. 1149.1 EXTEST instruction for non-HSSI pins.

If you use DC coupling on the HSSI signals, execute the EXTEST instruction. If you use AC coupling on the HSSI signals, execute the EXTEST_PULSE instruction.

I/O Voltage Support in a JTAG Chain

A device operating in BST mode uses four required pins—TDI, TDO, TMS, TCK, and one optional pin, TRST. The TCK pin has an internal weak pull-down resistor, while the TDI, TMS, and TRST pins have internal weak pull-up resistors. The TDO output pin and all the JTAG input pins are powered by the 2.5-V/3.0-V V_{CCPD} supply of I/O bank 3A. All user I/O pins are tri-stated during JTAG configuration.

The JTAG chain can support several different devices. However, use caution if the chain contains devices that have different V_{CCIO} levels. The output voltage level of the TDO pin must meet the specification of the TDI pin it drives.

Table 11–3 lists board design recommendations to ensure proper JTAG chain operation.

Table 11–3. Supported TDO and TDI Voltage Combinations (Part 1 of 2)

Device	TDI Input Buffer Power	Stratix V TDO V_{CCPD}	
		V_{CCPD} = 3.0 V (1)	V_{CCPD} = 2.5 V (2)
Stratix V	V_{CCPD} = 3.0 V	Y	Y
	V_{CCPD} = 2.5 V	Y	Y

Table 11–3. Supported TDO and TDI Voltage Combinations (Part 2 of 2)

Device	TDI Input Buffer Power	Stratix V TDO V_{CCPD}	
		V_{CCPD} = 3.0 V (1)	V_{CCPD} = 2.5 V (2)
Non-Stratix V	V_{CC} = 3.3 V	Y (3)	Y (4)
	V_{CC} = 2.5 V	Y (3)	Y (4)
	V_{CC} = 1.8 V	Y (3)	Y (4)
	V_{CC} = 1.5 V	Y (3)	Y (4)

Notes to Table 11–3:

(1) The TDO output buffer meets V_{OH} (MIN) = 2.4 V.
(2) The TDO output buffer meets V_{OH} (MIN) = 2.0 V.
(3) Input buffer must be 3.0-V tolerant.
(4) Input buffer must be 2.5-V tolerant.

Boundary-Scan Description Language Support

The boundary-scan description language (BSDL), a subset of VHDL, provides a syntax that allows you to describe the features of an IEEE Std. 1149.6 BST-capable device that can be tested. You can test software development systems, then use the BSDL files for test generation, analysis, and failure diagnostics.

For more information about BSDL files for IEEE Std. 1149.6-compliant Stratix V devices, refer to the IEEE 1149.6 BSDL Files page.

You can also generate BSDL files (pre-configuration and post-configuration) for IEEE Std. 1149.6-compliant Stratix V devices with the Quartus® II software version 10.0 SP1 and later. For more information about the procedure to generate BSDL files using the Quartus II software, refer to the BSDL Files Generation in QII page.

Document Revision History

Table 11–4 lists the revision history for this chapter.

Table 11–4. Document Revision History

Date	Version	Changes
June 2012	1.5	Updated Table 11–1.
December 2011	1.4	Updated Table 11–2 to include KEY_CLR_VREG and KEY_VERIFY JTAG instructions.
November 2011	1.3	Updated Table 11–1 and Table 11–2.
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 11–1.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

12. Power Management in Stratix V Devices

SV51013-1.3

This chapter describes power management in Stratix® V devices. Stratix V devices offer programmable power technology options for low-power operation. You can use these options, along with speed grade choices, in different permutations to give the best power and performance combination.

For thermal management, use the Stratix V internal temperature sensing diode (TSD) with built-in analog-to-digital converter (ADC) circuitry or external TSD with an external temperature sensor to easily incorporate this feature in your designs. Being able to monitor the junction temperature of the device at any time also allows you the ability to control air flow to the device and save power for the whole system.

Stratix V FPGAs deliver a breakthrough level of system bandwidth and power efficiency for high-end applications, allowing you to innovate without compromise. Stratix V devices use advanced power management techniques to enable both density and performance increases while simultaneously reducing power dissipation.

The total power of an FPGA includes static and dynamic power.

- Static power is the power consumed by the FPGA when it is configured but no clocks are operating.
- Dynamic power is comprised of switching power when the device is configured and running. You can calculate dynamic power with the equation shown in Equation 12–1.

Equation 12–1. Dynamic Power Equation *(1)*

$$P = \frac{1}{2} C V^2 \times \text{frequency}$$

Note to Equation 12–1:

(1) P = power; C = load capacitance; and V = supply voltage level.

Equation 12–1 shows that power is design-dependent and is determined by the operating frequency of the design. However, you can vary the voltage to lower dynamic power consumption by the square value of the voltage difference. Stratix V devices minimize static and dynamic power with advanced process optimizations and programmable power technology. These technologies enable Stratix V designs to optimally meet design-specific performance requirements with the lowest possible power.

The Quartus® II software optimizes all designs with Stratix V power technology to ensure performance is met at the lowest power consumption. This automatic process allows you to concentrate on the functionality of the design instead of the power consumption of the design.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Power consumption also affects thermal management. Stratix V devices offer a TSD feature that self-monitors the device junction temperature and can be used with external circuitry for other activities, such as controlling air flow to the Stratix V FPGA.

This chapter contains the following sections:

- "Stratix V Programmable Power Technology"
- "Stratix V External Power Supply Requirements"
- "Temperature Sensing Diode"

Stratix V Programmable Power Technology

Stratix V devices offer the ability to configure portions of the core, called tiles, for high-speed or low-power mode of operation performed by the Quartus II software without user intervention. Setting a tile to high-speed or low-power mode is accomplished with on-chip circuitry and does not require extra power supplies brought into the Stratix V device. In a design compilation, the Quartus II software determines whether a tile must be in high-speed or low-power mode based on the timing constraints of the design.

A Stratix V tile consists of the following:

- Memory logic array block (MLAB)/logic array block (LAB) pairs with routing to the pair
- MLAB/LAB pairs with routing to the pair and to adjacent digital signal processing (DSP)/memory block routing
- TriMatrix memory blocks
- DSP blocks
- PCI Express® (PCIe®) hard IP
- Physical Coding Sublayer (PCS)

All blocks and routing associated with the tile share the same setting of either high-speed or low-power mode. By default, tiles that include DSP blocks or memory blocks are set to high-speed mode for optimum performance. Unused DSP blocks and memory blocks are set to low-power mode to minimize static power. Clock networks do not support programmable power technology.

With programmable power technology, faster speed grade FPGAs may require less power because there are fewer high-speed MLAB and LAB pairs, when compared with slower speed grade FPGAs. The slower speed grade device may have to use more high-speed MLAB and LAB pairs to meet performance requirements, while the faster speed grade device can meet performance requirements with MLAB and LAB pairs in low-power mode.

The Quartus II software sets unused device resources in the design to low-power mode to reduce the static power. It also sets the following resources to low-power mode when they are not used in the design:

- LABs and MLABs
- TriMatrix memory blocks
- DSP blocks

If a phase-locked loop (PLL) is instantiated in the design, user may assert the `areset` pin high to keep the PLL in low-power mode.

Table 12–1 lists the available Stratix V programmable power capabilities. Speed grade considerations can add to the permutations to give you flexibility in designing your system.

Table 12–1. Programmable Power Capabilities for Stratix V Devices

Feature	Programmable Power Technology
LAB	Yes
Routing	Yes
Memory Blocks	Fixed setting (1)
DSP Blocks	Fixed setting (1)
Global Clock Networks	No

Note to Table 12–1:

(1) Tiles with DSP blocks and memory blocks that are used in the design are always set to high-speed mode. By default, unused DSP blocks and memory blocks are set to low-power mode.

Stratix V External Power Supply Requirements

This section describes the different external power supplies required to power Stratix V devices. You can supply some of the power supply pins with the same external power supply, provided they have the same voltage level requirements.

For more information about power supply pin connection guidelines and power regulator sharing, refer to the *Stratix V Device Family Pin Connection Guidelines*.

For each Altera-recommended power supply's operating conditions, refer to the *Stratix V Device Datasheet*.

Temperature Sensing Diode

The Stratix V TSD uses the characteristics of a PN junction diode to determine die temperature. Knowing the junction temperature is crucial for thermal management. Historically, junction temperature is calculated using ambient or case temperature, junction-to-ambient (ja) or junction to-case (jc) thermal resistance, and device power consumption. Stratix V devices can either monitor its die temperature with the internal TSD with built-in ADC circuitry or the external TSD with an external temperature sensor. This allows you to control the air flow to the device.

Internal Temperature Sensing Diode

You can use the Stratix V internal TSD in two different modes of operations—power-up mode and user mode. For power-up mode, the internal TSD reads the die's temperature during configuration if you enable the ALTTEMP_SENSE megafunction in your design. The ALTTEMP_SENSE megafunction allows temperature sensing during device user mode by asserting the `clken` signal to the internal TSD circuitry. To reduce power consumption, disable the internal TSD with built-in ADC circuitry when not in use.

Table 12–2 lists the internal TSD specification.

Table 12–2. Internal Temperature Sensing Diode Specification

Temperature Range	Accuracy	Offset Calibrated Option	Sampling Rate	Conversion Time	Resolution	Minimum Resolution with no Missing Codes
–40°C to 100°C	±8°C (1)	No	1 MHz, 500 KHz	< 100 ms	8 bits	8 bits

Note to Table 12–2:

(1) Pending on silicon characterization.

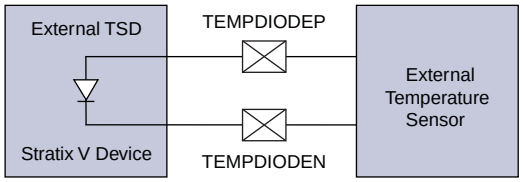
For more information about using the ALTTEMP_SENSE megafunction, refer to the *Thermal Sensor (ALTTEMP_SENSE) Megafunction User Guide*.

The external temperature sensor steers bias current through the Stratix V external TSD, which measures forward voltage and converts this reading to temperature in the form of an 8-bit signed number (7 bits plus sign). The 8-bit output represents the junction temperature of the Stratix V device and can be used for intelligent power management.

External Temperature Sensing Diode

The Stratix V external TSD requires two pins for voltage reference. Figure 12–1 shows how to connect the external TSD with an external temperature sensor device, allowing external sensing of the Stratix V die temperature. As an example, you can connect external temperature sensing devices, such as MAX1619, MAX1617A, MAX6627, and ADT 7411 to the two external TSD pins for Stratix V device die temperature reading.

Figure 12–1. TSD External Pin Connections for Stratix V Devices



For more information about the external TSD specification, refer to the *Stratix V Device Datasheet*.

The TSD is a very sensitive circuit that can be influenced by noise coupled from other traces on the board or within the device package itself, depending on your device usage. The interfacing signal from the Stratix V device to the external temperature sensor is based on millivolts (mV) of difference, as seen at the external TSD pins. Switching the I/O near the TSD pins can affect the temperature reading. Altera recommends taking temperature readings during periods of inactivity in the device or use the internal TSD with built-in ADC circuitry.

The following are board connection guidelines for the TSD external pin connections:

- The maximum trace lengths for the $TEMPDIODE_P$/$TEMPDIODE_N$ traces must be less than eight inches.
- Route both traces in parallel and place them close to each other with grounded guard tracks on each side.
- Altera recommends 10-mils width and space for both traces.
- Route traces through a minimum number of vias and crossunders to minimize the thermocouple effects.
- Ensure that the number of vias are the same on both traces.
- Ensure both traces are approximately the same length.
- Avoid coupling with toggling signals (for example, clocks and I/O) by having the GND plane between the diode traces and the high frequency signals.
- For high-frequency noise filtering, place an external capacitor (close to the external chip) between the $TEMPDIODE_P$/$TEMPDIODE_N$ trace.
  - For Maxim devices, use an external capacitor between 2200 pF to 3300 pF.
- Place a 0.1 uF bypass capacitor close to the external device.
- You can use internal TSD with built-in ADC circuitry and external TSD at the same time.
- If you only use internal ADC circuitry, the external TSD pins ($TEMPDIODE_P$/$TEMPDIODE_N$) can be connected to GND because the external TSD pins are not used.

  For more information about the $TEMPDIODE_P$/$TEMPDIODE_N$ pin connection when you are not using an external TSD, refer to the *Stratix V Device Family Pin Connection Guidelines*.

For more information about device specification and connection guidelines, refer to the external temperature sensor device data sheet from the device manufacturer.

Document Revision History

Table 12–3 shows the revision history for this chapter.

Table 12–3. Document Revision History

Date	Version	Changes
June 2012	1.3	Minor text edits.
May 2011	1.2	Chapter moved to volume 2 for the 11.0 release.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

This chapter provides additional information about the document and Altera.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Nontechnical support (general)	Email	nacomp@altera.com
(software licensing)	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *Stratix IV Design Guidelines*.
italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, *<file name>* and *<project name>***.pof** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, "Typographic Conventions."

Visual Cue	Meaning
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix n denotes an active-low signal. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
↵	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	The question mark directs you to a software help system with related information.
	The feet direct you to another document or website with related information.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.

Stratix V Device Handbook

Volume 2: Transceivers

101 Innovation Drive
San Jose, CA 95134
www.altera.com

SV5V2-1.7
12.0

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO
9001:2008
Registered

Contents

Chapter Revision Dates ix

Chapter 1. Transceiver Architecture in Stratix V Devices

Stratix V Device Layout 1–3
Stratix V GX/GT Channel and PCIe Hard IP (HIP) Variants Layout 1–3
GX Channel Variants 1–4
PCIe HIP Variants 1–4
PMA Architecture 1–5
Receiver Clock Data Recovery Unit 1–6
Lock-to-Reference Mode 1–6
Lock-to-Data Mode 1–7
Automatic Lock Mode 1–7
Receiver Buffer 1–8
Receiver Analog Settings 1–8
Programmable Differential On-Chip Termination (OCT) 1–9
Programmable V_{CM} 1–9
Signal Threshold Detection Circuitry 1–9
Continuous Time Linear Equalization (CTLE) 1–10
Decision Feedback Equalization (DFE) 1–11
EyeQ 1–11
Deserializer 1–11
Receiver PMA Bit-Slip 1–12
Transmitter PLLs 1–12
Bonded Configurations 1–14
Channel PLL Used as a CMU PLL (Transmitter PLL) 1–14
Auxiliary Transmit (ATX) PLL Architecture 1–15
Transmitter Buffer 1–16
Link Coupling 1–17
Programmable Transmitter Termination 1–17
Transmitter Analog Settings 1–17
Serializer 1–18
Transceiver Calibration Blocks 1–19
OCT Calibration 1–19
Offset Cancellation in the Receiver Buffer and Receiver CDR 1–19
ATX PLL Calibration 1–20
Standard PCS Architecture 1–20
Receiver Standard PCS Datapath 1–21
Word Aligner 1–22
Deskew FIFO 1–25
Rate Match (Clock Rate Compensation) FIFO 1–26
8B/10B Decoder 1–26
Byte Deserializer 1–27
Byte Ordering Block 1–28
Receiver Phase Compensation FIFO 1–30
Transmitter Standard PCS Datapath 1–30
Transmitter Phase Compensation FIFO 1–31
Byte Serializer 1–31
8B/10B Encoder 1–32

10G PCS Architecture . . . 1–33
Receiver 10G PCS Datapath . . . 1–35
Receiver Gearbox . . . 1–35
Block Synchronizer . . . 1–35
Disparity Checker . . . 1–36
Descrambler . . . 1–36
Frame Synchronizer . . . 1–36
Bit-Error Rate (BER) Monitor . . . 1–36
64B/66B Decoder . . . 1–37
CRC-32 Checker . . . 1–37
Receiver FIFO . . . 1–37
Transmitter 10G PCS Datapath . . . 1–38
Transmitter FIFO . . . 1–38
Frame Generator . . . 1–39
CRC-32 Generator . . . 1–39
64B/66B Encoder . . . 1–40
Scrambler . . . 1–40
Disparity Generator . . . 1–40
Transmitter Gearbox . . . 1–41
PCIe Gen 3 PCS Architecture . . . 1–42
Receiver PCIe Gen 3 PCS Datapath . . . 1–43
Block Sync . . . 1–43
Rate Match FIFO . . . 1–43
Decoder . . . 1–43
Descrambler . . . 1–44
Phase Compensation FIFO (Shared with Standard 8GB PCS) . . . 1–44
Transmitter PCIe Gen 3 PCS Datapath . . . 1–44
Phase Compensation FIFO (Shared with Standard PCS) . . . 1–44
Scrambler . . . 1–44
Encoder . . . 1–44
Gearbox . . . 1–44
PIPE Interface . . . 1–44
Auto Speed Negotiation Block . . . 1–45
Electrical Idle Inference Block . . . 1–45
Clock Data Recovery (CDR) Control Block . . . 1–45
Document Revision History . . . 1–45

Chapter 2. Transceiver Clocking in Stratix V Devices

Input Reference Clocking . . . 2–1
Input Reference Clock Sources . . . 2–2
Dedicated refclk Pins . . . 2–4
Dedicated refclk Pins Using the Reference Clock Network . . . 2–5
Fractional PLLs . . . 2–5
Internal Clocking . . . 2–5
Transmitter Clock Network . . . 2–6
Transmitter Clock Lines . . . 2–7
Clock Dividers . . . 2–12
Transmitter Clock Network in Stratix V GT Transceiver Channels . . . 2–12
Transmitter Clocking . . . 2–13
Transmitter 10G PCS Clocking . . . 2–14
Transmitter Standard PCS Clocking . . . 2–15
Transmitter GT Channel Clocking . . . 2–21
Receiver Clocking . . . 2–21
GX Channel Receiver Clocking . . . 2–22

GT Channel Receiver Clocking . . . 2–29
FPGA Fabric-Transceiver Interface Clocking . . . 2–29
Transmitter Datapath Interface Clocking . . . 2–31
Quartus II-Selected Transmitter Datapath Interface Clock . . . 2–32
User-Selected Transmitter Datapath Interface Clock . . . 2–33
Receiver Datapath Interface Clock . . . 2–35
Quartus II Software-Selected Receiver Datapath Interface Clock . . . 2–36
User-Selected Receiver Datapath Interface Clock . . . 2–37
GXB 0 PPM Core Clock Assignment . . . 2–39
Document Revision History . . . 2–40

Chapter 3. Transceiver Reset Control in Stratix V Devices

Transceiver Reset Sequence . . . 3–1
Transceiver Blocks Affected by the Reset and Powerdown Signals . . . 3–2
PHY IP Embedded Reset Controller . . . 3–3
Timing Diagram . . . 3–5
User-Controlled Reset Controller . . . 3–6
Timing Diagram During Device Power-Up and After Reconfiguration . . . 3–7
CDR Modes During the Reset Sequence . . . 3–9
Reset Sequence for CDR in Manual Lock Mode . . . 3–9
Document Revision History . . . 3–11

Chapter 4. Transceiver Configurations in Stratix V Devices

10GBASE-R . . . 4–2
Transceiver Datapath Configuration . . . 4–4
Supported Features . . . 4–5
64-Bit Single Data Rate (SDR) Interface to the MAC/RS . . . 4–5
64B/66B Encoding/Decoding . . . 4–6
Transmitter and Receiver State Machines . . . 4–6
Block Synchronizer . . . 4–7
Self-Synchronous Scrambling/Descrambling . . . 4–7
BER Monitor . . . 4–7
Clock Compensation . . . 4–7
Transceiver Clocking . . . 4–8
Interlaken . . . 4–8
Transceiver Datapath Configuration . . . 4–8
Supported Features . . . 4–10
Block Synchronizer . . . 4–11
64B/67B Frame Generator . . . 4–11
Frame Synchronizer . . . 4–11
Running Disparity . . . 4–12
Frame Synchronous Scrambling/Descrambling . . . 4–12
Skip Word Insertion . . . 4–12
Skip Word Deletion . . . 4–12
Diagnostic Word Generation and Checking of Lane Data Integrity (CRC-32) . . . 4–12
Framing Layer Control Word Forwarding . . . 4–12
Multi-Lane Deskew Alignment . . . 4–13
Transmit and Receive FIFO Control and Status . . . 4–13
Transceiver Multi-Lane Bonding and Transmit Skew . . . 4–13
Transceiver Clocking . . . 4–13
Transceiver Clocking . . . 4–13
PCI Express (PCIe)—Gen1, Gen2, and Gen3 . . . 4–15
Transceiver Datapath Configuration . . . 4–16

Transceiver Channel Datapath 4–18
Supported Features for PCIe Configurations 4–19
PIPE 2.0 Interface 4–19
Dynamic Switching Between Gen1 (2.5 Gbps) and Gen2 (5 Gbps) Signal Rates 4–20
Transmitter Electrical Idle Generation 4–20
Power State Management 4–20
8B/10B Encoder Usage for Compliance Pattern Transmission Support 4–21
Receiver Electrical Idle Inference 4–21
Receiver Status 4–21
Receiver Detection 4–21
Clock Rate Compensation Up to ±300 PPM 4–22
PCIe Reverse Parallel Loopback 4–22
Supported Features for PCIe Gen3 4–23
Block Synchronization (Word Aligner) 4–23
Rate Match/Clock Compensation FIFO 4–23
Encoder/Decoder 4–23
Gen3 Gear Box 4–24
Scrambler/Descrambler 4–24
PIPE Gen3 Interface 4–24
Auto-Speed Negotiation Block 4–24
Transmitter Electrical IDLE Generation 4–25
Receiver Electrical IDLE Inference 4–25
Gen3 Power State Management 4–26
CDR Control Block 4–26
Transceiver Clocking and Channel Placement Guidelines 4–26
Transceiver Clocking for PCIe Gen1 and Gen2 4–26
Transceiver Channel Placement Guidelines for Gen1 and Gen2 PIPE Configurations 4–33
Transceiver Clocking for PCIe Gen3 (Hard IP) 4–35
Hard IP x1 Configuration 4–36
Hard IP x4 Configuration 4–37
Hard IP x8 Configuration 4–40
CPRI and OBSAI—Deterministic Latency Protocols 4–40
Transceiver Datapath Configuration 4–40
Receiver Phase Compensation FIFO in Register Mode 4–42
Transmitter Phase Compensation FIFO in Register Mode 4–42
Channel PLL Feedback 4–42
CPRI and OBSAI 4–43
XAUI 4–46
Transceiver Datapath in a XAUI Configuration 4–47
Supported Features 4–48
64-Bit SDR Interface to the MAC/RS 4–48
8B/10B Encoding/Decoding 4–49
Transmitter and Receiver State Machines 4–50
Synchronization 4–50
Deskew 4–50
Clock Compensation 4–50
Transceiver Clocking and Channel Placement Guidelines 4–51
Transceiver Clocking 4–51
Transceiver Channel Placement Guidelines 4–52
Stratix V Device Configurations 4–54
Standard PCS Configurations 4–54
Custom Datapath 4–55
Low Latency Datapath 4–61
Transceiver Channel Placement Guidelines 4–67

10G PCS Configurations 4–69
Datapath Functionality 4–72
Using the coreclkin Ports 4–77
Merging Instances 4–77
Transceiver Channel Placement Guidelines 4–77
Stratix V GX Native PHY IP 4–78
Stratix V GT Device Configurations 4–79
Document Revision History 4–80

Chapter 5. Transceiver Loopback Support in Stratix V Devices
Serial Loopback 5–1
PCIe Reverse Parallel Loopback 5–2
Reverse Serial Loopback 5–3
Reverse Serial Pre-CDR Loopback 5–4
Document Revision History 5–5

Chapter 6. Dynamic Reconfiguration in Stratix V Devices
Offset Cancellation Calibration 6–2
PMA Controls Reconfiguration 6–2
On-Chip Signal Quality Monitoring (EyeQ) 6–2
Decision Feedback Equalization 6–3
Adaptive Equalization 6–3
Enabling and Disabling Loopback Modes 6–4
Transceiver PLL Reconfiguration 6–4
Transceiver Channel and Interface Reconfiguration 6–5
Transceiver Channel Reconfiguration 6–5
Transceiver Interface Reconfiguration 6–5
Document Revision History 6–6

Additional Information
How to Contact Altera Info–1
Typographic Conventions Info–1

Chapter Revision Dates

The chapters in this document, *Stratix V Device Handbook Volume 2: Transceivers*, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Transceiver Architecture in Stratix V Devices
Revised: *June 2012*
Part Number: *SV52002-2.3*

Chapter 2. Transceiver Clocking in Stratix V Devices
Revised: *June 2012*
Part Number: *SV52003-1.6*

Chapter 3. Transceiver Reset Control in Stratix V Devices
Revised: *June 2012*
Part Number: *SV52004-3.3*

Chapter 4. Transceiver Configurations in Stratix V Devices
Revised: *June 2012*
Part Number: *SV52005-2.3*

Chapter 5. Transceiver Loopback Support in Stratix V Devices
Revised: *June 2012*
Part Number: *SV52007-2.4*

Chapter 6. Dynamic Reconfiguration in Stratix V Devices
Revised: *June 2012*
Part Number: *SV52008-2.3*

1. Transceiver Architecture in Stratix V Devices

SV52002-2.3

This chapter provides an overview of the Stratix® V GT, GX, and GS transceiver architecture, transceiver channels, and a description of the transmitter (TX) and receiver (RX) channel datapaths.

For a complete understanding of Stratix V transceivers, first review this overview of the transceiver architecture, then refer to the subsequent chapters in this volume for more detailed information.

For information about using Altera's transceiver IP, refer to the *Altera Transceiver PHY IP Core User Guide*.

Stratix V devices provide up to 66 backplane-capable full-duplex clock data recovery (CDR)–based transceivers with physical coding sublayer (PCS) and physical medium attachment (PMA). Stratix V devices support backplane capabilities up to 14.1 gigabits per second (Gbps).

- Stratix V GX and GS devices contain GX channels and have serial data rates that range from 600 megabits per second (Mbps) to 14.1 Gbps.
- Stratix V GT devices contain GT channels that have serial data rates that range from 19.6 Gbps to 28.05 Gbps. Stratix V GT devices also contain GX channels with a maximum data rate of up to 12.5 Gbps.

Transceiver performance over backplanes depends on the backplanes' characteristics.

Stratix V transceivers are divided into two blocks: PMA and PCS. The PMA block connects the FPGA to the channel, generates the required clocks, and converts the data from parallel to serial or serial to parallel. The PCS block performs digital processing logic between the PMA and the FPGA core. The PCS includes the encoder/decoders, scramblers/descramblers, word alignment, and rate matching FIFO blocks. There are three types of PCS blocks in Stratix V devices: a standard PCS block that supports data rates up to 11 Gbps, a 10G PCS block that supports data rates up to 14.1 Gbps, and a PCIe Gen3 PCS that supports the PCIe Gen 3 Base specification.

The following sections are included in this chapter:

- "Stratix V Device Layout" on page 1–3
- "PMA Architecture" on page 1–5
- "Standard PCS Architecture" on page 1–20
- "10G PCS Architecture" on page 1–33
- "PCIe Gen 3 PCS Architecture" on page 1–42

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

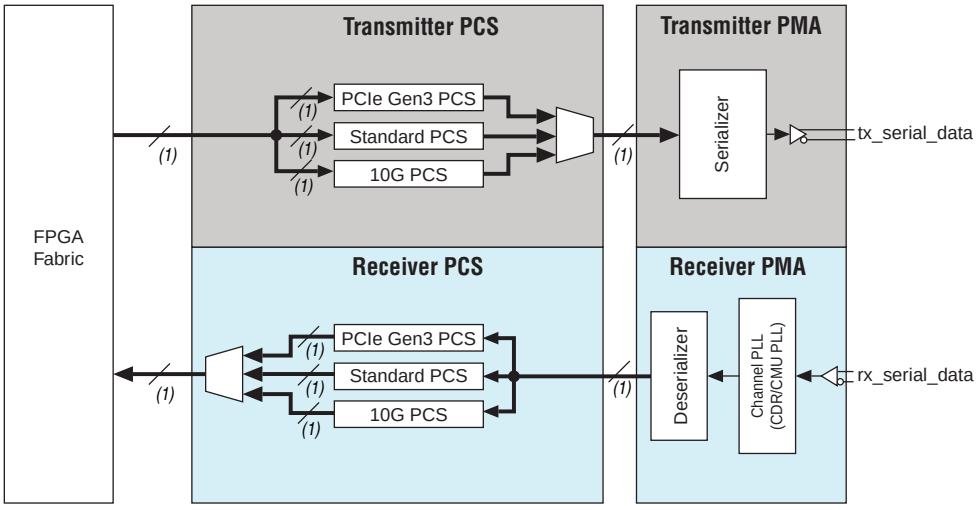
ISO 9001:2008 Registered

■ "Document Revision History" on page 1–45

For information about features that will be supported in a future release of the Quartus® II software, refer to the *Upcoming Stratix V Device Features* document.

Stratix V GT, GX, and GS transceivers are structured into full-duplex (transmitter and receiver) six-channel groups called transceiver blocks, shown in Figure 1–1 and Figure 1–2. The transmitter and receiver can operate separately.

Figure 1–1. Full-Duplex Channel Showing PMA and PCS Interfaces for the GX Channels

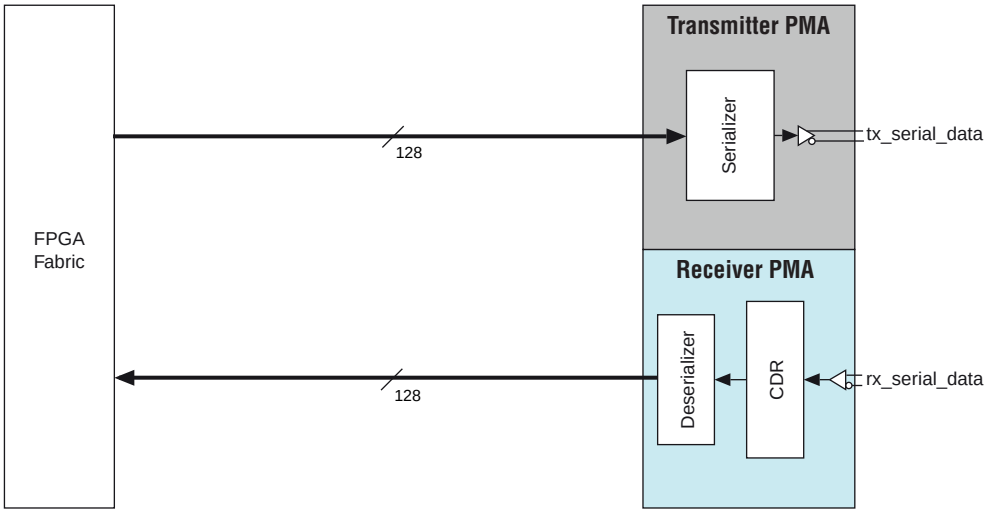


Note to Figure 1–1:

(1) The PMA and PCS widths are configurable.

The GT channels do not have PCS.

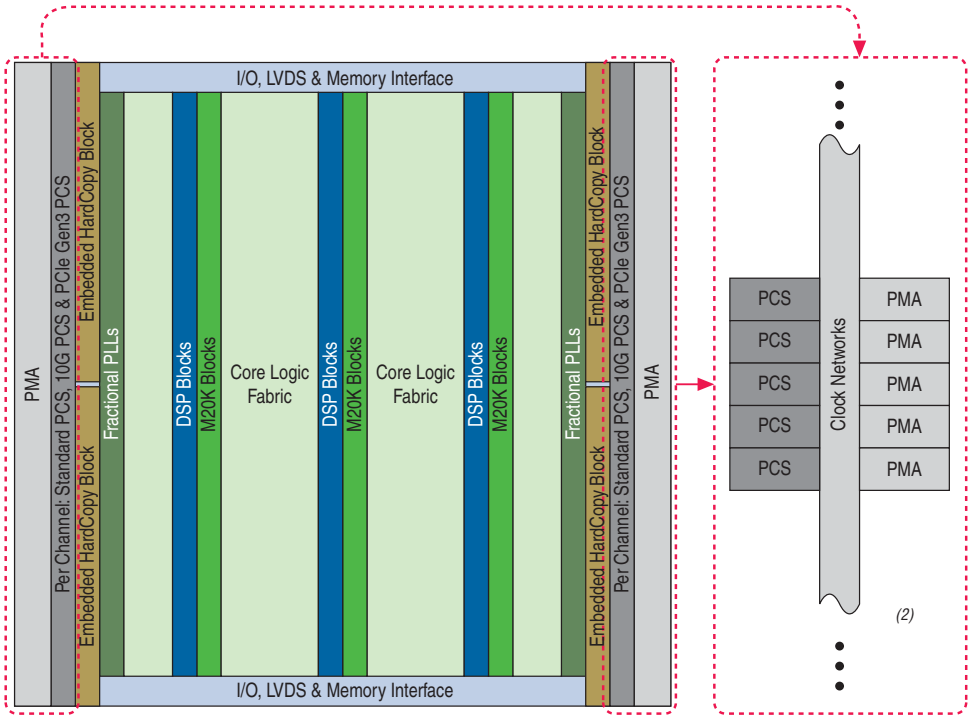
Figure 1–2. Full-Duplex Channel Showing the PMA Interface for the GT Channels



Stratix V Device Layout

Stratix V devices have columns of transceivers on the left and right sides of the devices. However, some Stratix V devices have columns of transceivers on the left side of the devices only. Figure 1–3 shows a basic layout of transceivers in a Stratix V device.

Figure 1–3. Basic Layout of Transceivers in a Stratix V Device *(1)*



Notes to Figure 1–3:

(1) This figure represents a given variant of a Stratix V device with transceivers. Other variants may have a different floor plan than the one shown here.

(2) You can use the unused transceiver channels as additional transceiver transmitter PLLs.

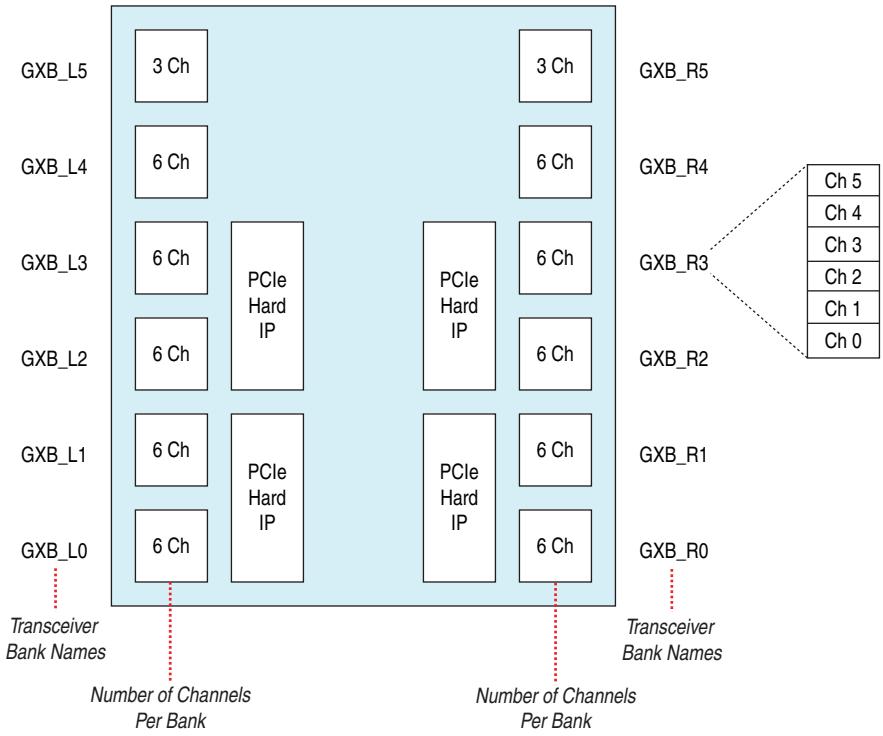
The location of the transceiver bank boundaries are important for clocking resources, bonding channels, and fitting. The transceivers are grouped in transceiver banks of three and six channels. Figure 1–4 on page 1–4 through Figure 1–5 on page 1–5 show the location of the transceiver blocks. Some package variations reduce the transceiver count in Figure 1–4 on page 1–4 and Figure 1–5 on page 1–5.

Within a transceiver bank the lowest numbered pin name is channel 0 of that bank and the highest numbered pin name of the bank is channel 5.

Stratix V GX/GT Channel and PCIe Hard IP (HIP) Variants Layout

Stratix V devices have many different channel and PCIe Hard IP variants. Figure 1–4 and Figure 1–5 show the general layout for these variants.

Figure 1–4. General Transceiver Bank locations for Stratix V GX Devices *(1)*



Note for Figure 1–4:

(1) Figure 1–1 shows the layout for a 66 channel device. Some devices have less channels.

GX Channel Variants

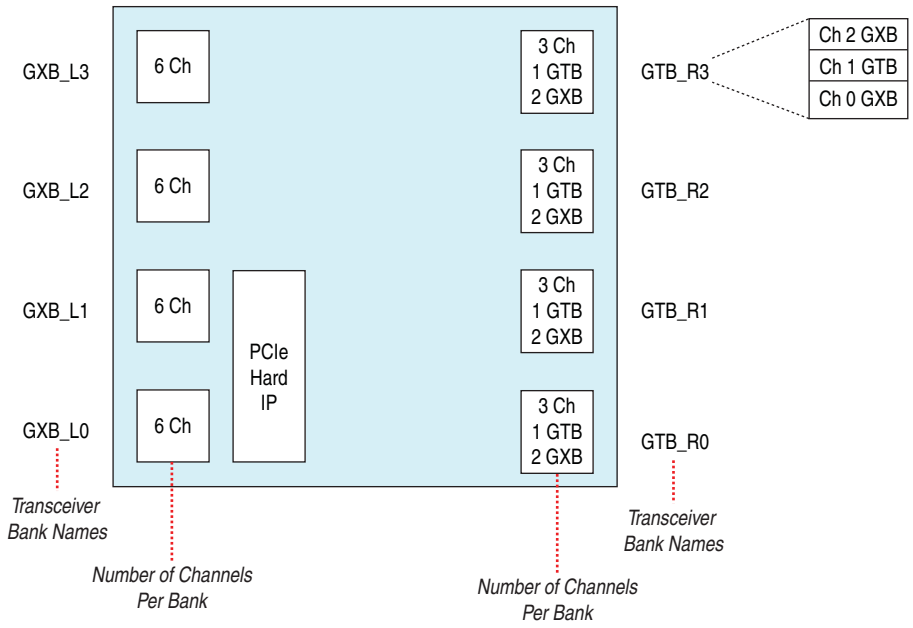
- 12-channel devices use banks L0 and L1
- 24-channel devices use banks L0, L1, R0 and R1
- 36-channel devices use banks L0, L1, L2, R0, R1 and R2
- 48-channel devices use banks L0, L1, L2, L3, R0, R1, R2 and R3
- 66-channel devices use banks uses all banks as shown in Figure 1–4

PCIe HIP Variants

- 1 PCIe HIP variant has an HIP block located across GX banks L0 and L1
- 2 PCIe HIP variant has HIP blocks located across GX banks L0 and L1 and GX banks R0 and R1
- 4 PCIe HIP variant has HIP blocks located across GX banks L0 and L1, GX banks L2 and L3, GX banks R0 and R1 and GX banks R2 and R3

Figure 1–5 shows the location of the transceiver blocks for the Stratix V GT devices.

Figure 1–5. Transceiver Bank locations for Stratix V GT Devices



Notes for Figure 1–5:

(1) GT transceiver banks are made up of 1 GT channel and 2 GX channels. The GT channel is the middle channel in the bank.

(2) GT devices only come with one PCIe HIP block located across GX banks L0 and L1.

For information about device pin-outs, refer to Pin-Out Files for Altera Devices.

For information about device options, refer to the *Stratix V Device Overview*.

PMA Architecture

The PMA receives and transmits off-chip high-speed serial data streams. There are two types of PMAs in Stratix V devices: GX channel PMAs and GT channel PMAs. The GT channel PMAs operate up to 28.05 Gbps and are only available in Stratix V GT devices. The GX channel PMAs are available in Stratix V GT, GX, and GS devices and either operate up to 14.1 Gbps (GX and GS devices) or 12.5 Gbps (GT devices).

The following sections describe the Stratix V PMA architecture:

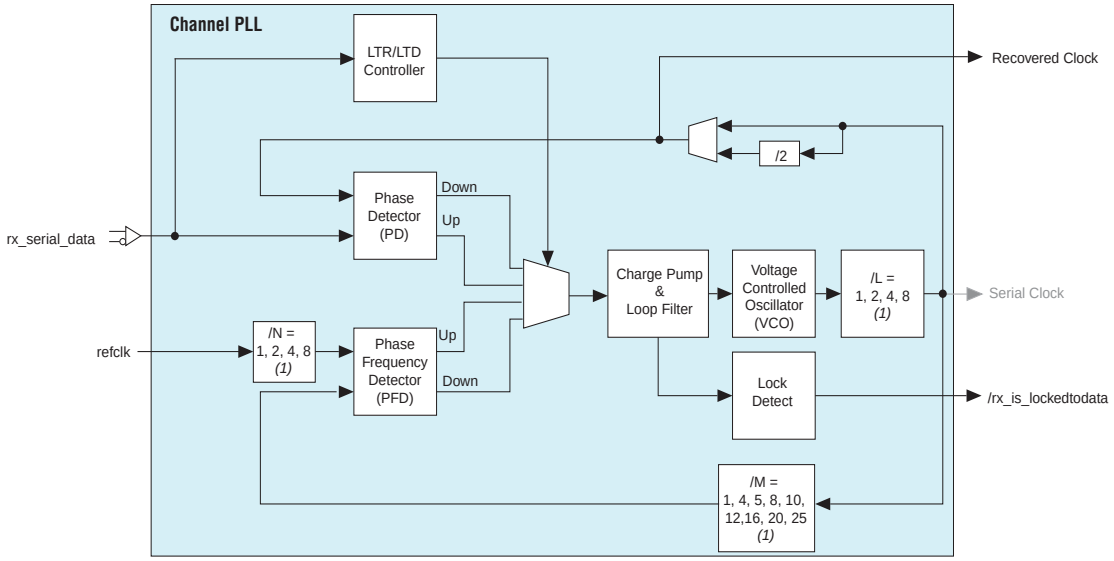
- "Receiver Clock Data Recovery Unit"
- "Receiver Buffer" on page 1–8
- "Transmitter PLLs" on page 1–12
- "Transmitter Buffer" on page 1–16
- "Transceiver Calibration Blocks" on page 1–19

Receiver Clock Data Recovery Unit

The PMA of each receiver channel includes a channel PLL that you can configure as receiver clock data recovery (CDR) PLL, or a clock multiplier unit (CMU) PLL. For more information about using the channel PLL as a CMU PLL, refer to "Transmitter PLLs" on page 1–12.

Figure 1–6 shows the portion of a channel PLL that functions as a CDR PLL.

Figure 1–6. Channel PLL Configured as CDR PLL in Stratix V Devices



Note to Figure 1–6:

(1) Not all combinations of /N, /M, and/L values are valid. The Quartus II software automatically chooses the optimal values.

The CDR supports the full range of data rates. The voltage-controlled oscillator (VCO) operates at half rate. The settings are automatically chosen by the Quartus II software.

The CDR operates in either lock-to-reference (LTR) or lock-to-data (LTD) mode. When the CDR is locked to the input reference clock, the CDR output clock is trained to the configured data rate. The CDR next switches to LTD mode to recover the clock from the incoming data. The LTR/LTD controller determines which mode the CDR is in, either manually, or automatically.

Lock-to-Reference Mode

In LTR mode, the phase frequency detector (PFD) in the CDR tracks the receiver input reference clock. The PFD controls the charge pump that tunes the VCO in the CDR. The `pma_rx_is_lockedtoref` status signal is asserted active high to indicate that the CDR has locked to the phase and frequency of the receiver input reference clock.

The phase detector (PD) is inactive in LTR mode and `pma_rx_is_lockedtodata` is ignored.

Lock-to-Data Mode

During normal operation, the CDR must be in LTD mode to recover the clock from the incoming serial data. In LTD mode, the PD in the CDR tracks the incoming serial data at the receiver buffer. Depending on the phase difference between the incoming data and the CDR output clock, the PD controls the CDR charge pump that tunes the VCO.

The PFD is inactive in LTD mode. The `rx_is_lockedtoref` status signal toggles randomly and is not significant in LTD mode.

After switching to LTD mode, the `rx_is_lockedtodata` status signal is asserted. Lock to data takes a minimum of 4 µs, however the actual lock time depends on the transition density of the incoming data and the parts per million (ppm) difference between the receiver input reference clock and the upstream transmitter reference clock. The receiver PCS logic must be held in reset until the CDR produces a stable recovered clock.

For more information about status signals, refer to the *Stratix V Device Datasheet*.

Automatic Lock Mode

In automatic lock mode, the LTR/LTD controller initially sets the CDR to lock to the input reference clock (LTR mode). After the CDR locks to the input reference clock, the LTR/LTD controller automatically sets the CDR to lock to the incoming serial data (LTD mode) when the following conditions are met:

- The signal threshold detection circuitry indicates the presence of valid signal levels at the receiver input buffer.
- The CDR output clock is within the configured ppm frequency threshold setting with respect to the input reference clock (frequency locked).
- The CDR output clock and the input reference clock are phase matched within approximately 0.08 unit interval (UI) (phase locked).

The switch back and forth from LTR to LTD mode, is indicated by the assertion of the `pma_rx_is_lockedtodata` signal.

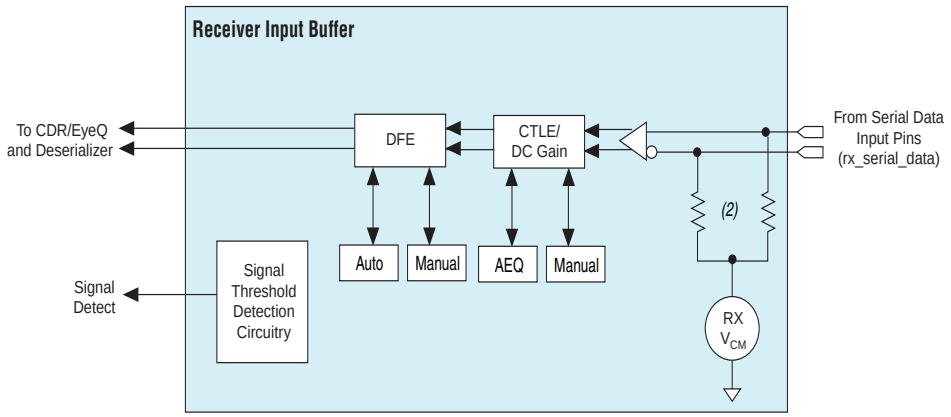
If the CDR does not stay locked to data because of frequency drift or severe amplitude attenuation, the LTR/LTD controller switches the CDR back to LTR mode to lock to the input reference clock. In automatic lock mode, the LTR/LTD controller switches the CDR from LTD mode to LTR mode when the following conditions are met:

- The signal threshold detection circuitry indicates the absence of valid signal levels at the receiver input buffer for PCI Express® (PCIe®) configurations only. This condition defaults to **true** for all other configurations.
- The CDR output clock is not within the configured ppm frequency threshold setting with respect to the input reference clock.

Receiver Buffer

The receiver input buffer receives serial data from the rx_serial_data port and feeds the serial data to the channel PLL configured as a CDR PLL, as shown in Figure 1–7.

Figure 1–7. Receiver Input Buffer *(1)*



Notes to Figure 1–7:

(1) Stratix V GT channels do not support AEQ and DFE.

(2) For Stratix V GX channels, the available options are OFF, 85, 100, 120, and 150 Ω. Stratix V GT channels only supports 100 Ω.

Modifying programmable values within receiver input buffers can be performed by a single reconfiguration controller for the entire FPGA, or multiple reconfiguration controllers if desired. Within each transceiver bank a maximum of two reconfiguration controllers is allowed; one for the three-transceiver triplet in the upper-half of the bank, and one for the lower-half. This is due to a single slave interface to all PLLs and PMAs within each triplet. Therefore, many triplets can be connected to a single reconfiguration controller, but only one reconfiguration controller can be connected to the three transceivers within any triplet.

A maximum of one reconfiguration controller is allowed per transceiver bank upper-half or lower-half triplet.

For more information about the Receiver Buffer, refer to the *Altera Transceiver PHY IP Core User Guide*.

For information about the Stratix V electrical specifications, refer to the *Stratix V Device Datasheet*.

Receiver Analog Settings

Stratix V GX channels have two receiver analog modes: half-bandwidth and full-bandwidth. The half-bandwidth data rate is up to 6.25 Gbps; the full-bandwidth data rate is from 6.25 Gbps to 14.1 Gbps. You can select which mode to use in the Assignment Editor of the Quartus II software (Receiver Equalizer Gain Bandwidth Select).

Stratix V GT channels have only one receiver analog mode.

For more information about receiver analog performance, refer to the *Stratix V Device Datasheet*.

Programmable Differential On-Chip Termination (OCT)

The receiver buffer supports optional differential OCT resistances of 85, 100, 120, 150 Ω and OFF for GX channels. The receiver buffer of GT channels only supports 100 Ω differential termination but provides for fine adjustment.

The receiver OCT resistors have calibration support to compensate for process, voltage, and temperature (PVT) variations.

Programmable V_{CM}

The receiver buffer has on-chip biasing circuitry to establish the required V_{CM} at the receiver input.

The Quartus II software automatically chooses the best V_{CM} setting. For more information, refer to the *Stratix V Device Datasheet*.

On-chip biasing circuitry is available only if you select one of the **Termination logic options** to configure OCT. If you select external termination, you must implement off-chip biasing circuitry to establish the V_{CM} at the receiver input buffer.

Signal Threshold Detection Circuitry

In a PCIe configuration for GX channels, you can enable the optional signal threshold detection circuitry. If enabled, this option senses whether the signal level present at the receiver input buffer is above the signal detect threshold voltage that you specified. This option is not available for GT channels.

For more information about the signal threshold detect signal, refer to the *Altera Transceiver PHY IP Core User Guide*.

DC Gain

DC gain circuitry provides an equal boost to the incoming signal across the frequency spectrum. The receiver buffer supports DC gain settings of up to 8 dB.

For more information about the DC gain settings for the receiver buffer, refer to the *Stratix V Device Datasheet*.

Continuous Time Linear Equalization (CTLE)

Continuous time linear equalization is available in Stratix V receivers. Each receiver buffer has five independently programmable equalization circuits that boost the high-frequency gain of the incoming signal, thereby compensating for the low-pass characteristics of the physical medium. For Stratix V GX channels, the equalization circuitry provides up to 16 dB of high-frequency boost. For Stratix V GT channels, the equalization circuitry provides up to 15 dB of high-frequency boost. The CTLE operates in two modes: manual mode and adaptive equalization (AEQ) mode.

For more information about the CTLE specifications, refer to the *Stratix V Device Datasheet*.

Manual Mode

Manual mode allows you to manually adjust the continuous time linear equalization to improve signal integrity. You can statically set the equalizer settings in the IP or you can dynamically change the equalizer settings with the reconfiguration controller IP.

Adaptive Equalization Mode

AEQ mode eliminates the need for manual tuning by enabling the Stratix V device to automatically tune the receiver equalization settings based on the frequency content of the incoming signal and comparing that with internally generated reference signals. The AEQ block resides within the PMA of the receiver channel and is available on all GX channels.

AEQ mode is not supported in GT channels.

There are three AEQ modes: continuous, one-time, and powerdown:

- Continuous mode—The AEQ continuously monitors the frequency content of the received signal and adapts to the received signal by providing dynamically changing equalizer settings to the receiver.
- One-time mode—The AEQ finds a stable setting of the receiver equalizer and locks to that value. After the stable setting is locked, the equalizer values do not.
- Powerdown mode—The AEQ of the specific channel is placed in standby mode and the CTLE uses the manually set value. Note that the CTLE cannot be bypassed.

You can dynamically switch between these modes.

For more information about enabling different options and using them to control the AEQ hardware, refer to the "AEQ" section in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

Decision Feedback Equalization (DFE)

The decision feedback equalization feature consists of a 5-tap equalizer, which boosts the high frequency components of a signal without noise amplification by compensating for inter-symbol interference (ISI). There are two decision feedback equalization modes: manual and auto-adaptation.

For more information, refer to the "DFE" section in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

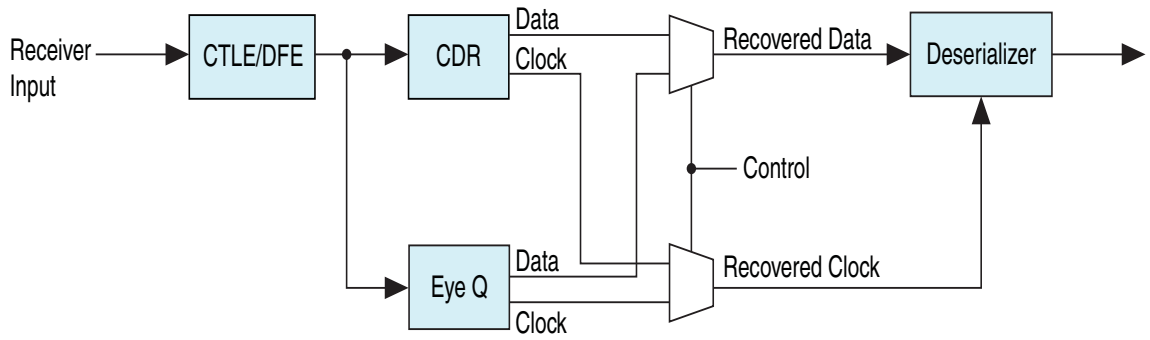
The DFE is not supported in GT channels.

EyeQ

The EyeQ feature is a debug and diagnosis tool that helps you analyze the received data by measuring the horizontal and vertical eye opening.

Figure 1–8 shows the receiver and EyeQ architecture. There are two multiplexers for the data and clock which select one path to feed to the deserializer.

Figure 1–8. Receiver and EyeQ Architecture



For more information about the EyeQ feature, refer to the *Altera Transceiver PHY IP Core User Guide*.

Deserializer

The deserializer block clocks in serial input data from the receiver buffer using the high-speed serial recovered clock and deserializes the data using the low-speed parallel recovered clock. The deserializer forwards the deserialized data to the receiver PCS or FPGA core.

The GX channel deserializer supports factors of 8, 10, 16, 20, 32, and 40 bits. Unlike the GX channel deserializer, the GT channel deserializer does not support programmable data widths and is fixed at 128 bits.

Receiver PMA Bit-Slip

The deserializer has a bit slip feature to enable the high speed serial bit slipping to minimize uncertainty in serialization process per CPRI requirement. It is controlled by the PCS in Custom and deterministic latency modes configured from CPRI. The period of a parallel clock (clkdiv_rx) could be extended by 1 UI during clock slip operation.

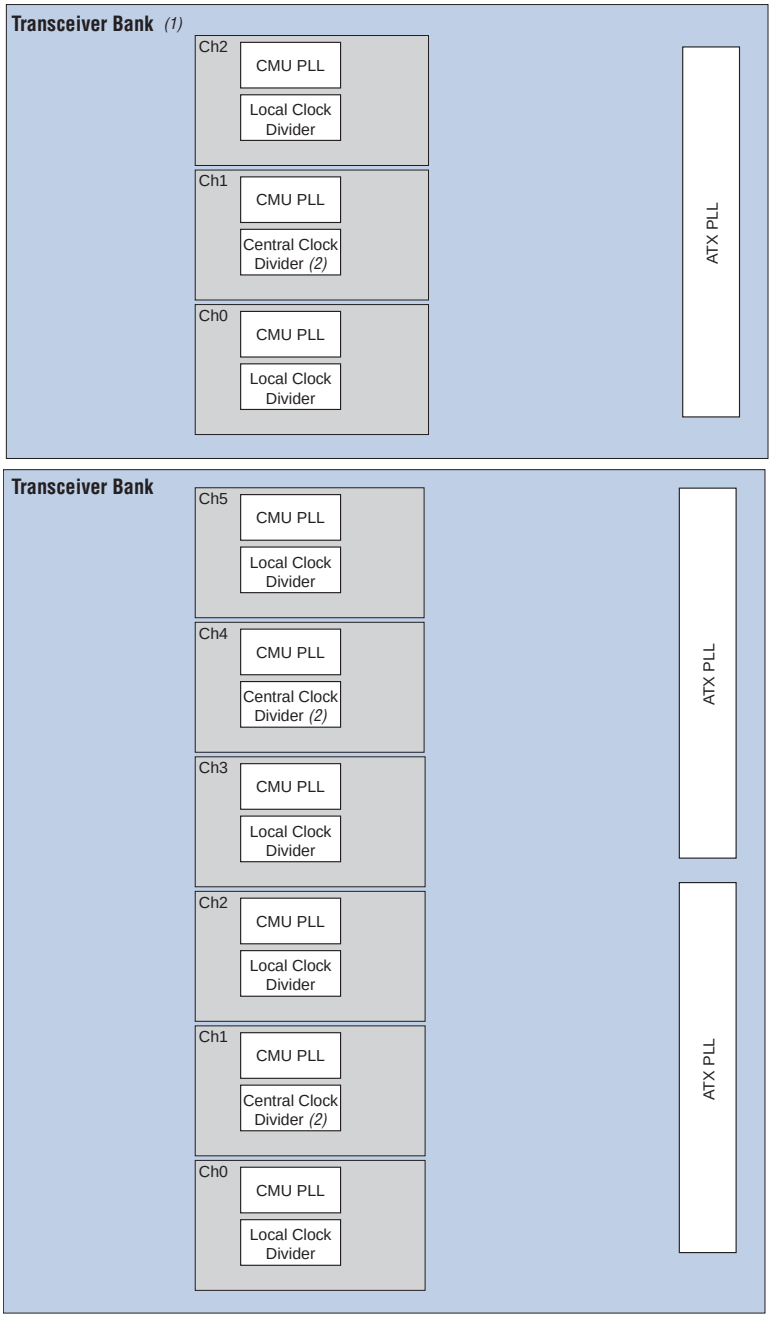
For more information, refer to the *Altera Transceiver PHY IP Core User Guide*.

Transmitter PLLs

The following sections describe the Stratix V transmitter PLLs. Figure 1–9 shows the transmitter PLL locations and clocking for Stratix V GX channels.

For more information about clocking, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

Figure 1–9. Transmitter PLL Locations in Stratix V GX Channels



Notes to Figure 1–9:

(1) Stratix V devices 5SGXB5, 5SGXB6, 5SGSB9 and 5SGSBB have one transceiver bank on each side with only three transceiver channels.

(2) You can use the central clock divider as a local clock divider.

Stratix V GT channels must use auxiliary transmit (ATX) PLLs.

Transmitter PLLs within the upper-half or lower-half of a transceiver bank must be connected to the same Reconfiguration Controller.

Each transmitter channel has a clock divider called a local clock divider. The clock dividers generate the parallel and serial clock sources for the transmitter and optionally for the receiver PCS. The clock dividers on channels 1 and 4 are called central clock dividers as they can drive the x6 and xN clock lines. The central clock dividers can feed the clock lines used to bond channels.

For more information about the clock divider, refer to the *Clock Dividers* section in the *Transceiver Clocking in Stratix V Devices* chapter.

Bonded Configurations

The high-speed serial clock and low-speed parallel clock skew between channels and unequal latency in the transmitter phase compensation FIFO contribute to transmitter channel-to-channel skew. In bonded channel configurations the parallel clock is generated by a central clock divider for all channels, rather than using a local clock divider for each transmitter channel. Also, the transmitter phase compensation FIFO in all bonded channels shares common pointers and control logic generated in the central clock divider, resulting in equal latency in the transmitter phase compensation FIFO of all bonded channels. The lower transceiver clock skew and equal latency in the transmitter phase compensation FIFOs in all channels provide lower channel-to-channel skew in bonded channel configurations.

For more information about bonding, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

Channel PLL Used as a CMU PLL (Transmitter PLL)

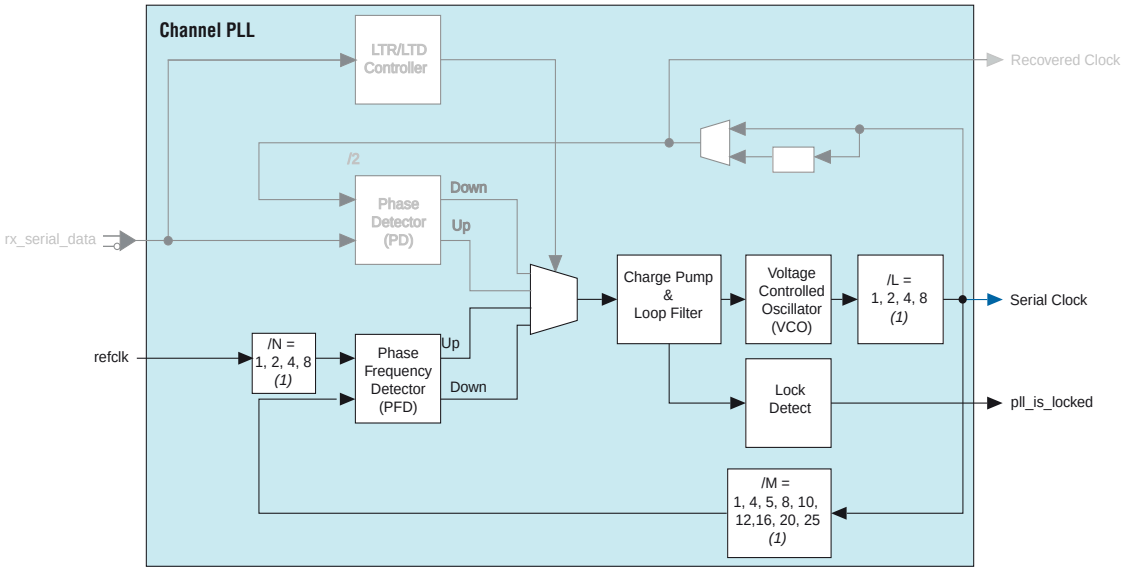
The receiver of every GX transceiver has a channel PLL that can function as a CMU PLL or a CDR PLL. This section describes the CMU PLLs. The CMU PLLs in channels 1 and 4 in a transceiver block can also provide a clock to the other transceivers within the same block or to another transceiver block.

When you use the channel PLL as a CMU PLL, the transmitter and receiver are not available. If all transmitters and receivers within the transceiver block are required, you must use an ATX PLL or a clock from another transceiver block.

The serial clock from the CMU PLL is routed to the transmitter clock dividers and can be further divided down to half the data rate of the individual channels. For the best performance based on the data rate and input clock frequency, all settings for the CMU PLL and clock dividers are automatically chosen by the Quartus II software.

Figure 1–10 shows the portion of a channel PLL that functions as a CMU PLL.

Figure 1–10. Channel PLL Configured as CMU PLL in Stratix V Devices



Note to Figure 1–10:

(1) Not all combinations of /N, /M, and/L values are valid. The Quartus II software automatically chooses the optimal values.

For CMU PLL specifications such as input clock frequency or supported output data ranges, refer to the *Stratix V Device Datasheet*.

Transmitter PLLs within the upper-half or lower-half of a transceiver bank must be connected to the same Reconfiguration Controller.

Auxiliary Transmit (ATX) PLL Architecture

Most Stratix V GT, GX, and GS devices contain two ATX PLLs per transceiver bank that can generate the high-speed clocks for the transmitter channels; the 66 channel device is an exception with only one ATX PLL channel in the top bank. Compared with CMU PLLs, ATX PLLs have lower jitter and do not consume a transceiver channel; however an ATX PLL's frequency range is more limited.

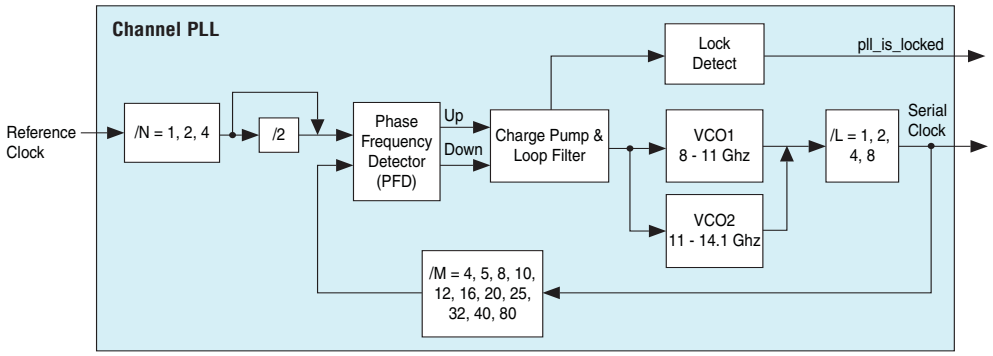
The serial clock from the ATX PLL is routed to the transmitter clock dividers and can be further divided down to half the data rate of the individual channels. For best performance you should use the reference clock input that resides in the same transceiver block as your channel. However, you can use any dedicated reference clocks along the same side of the device to clock the ATX PLL.

For ATX PLL specifications such as input clock frequency or supported output data ranges, refer to the *Stratix V Device Datasheet*.

All Stratix V GT channels must use ATX PLLs. The reference clock input should reside in the same transceiver block as the channel for best performance.

Figure 1–11 shows the ATX PLL architecture.

Figure 1–11. ATX PLL Architecture



Transmitter Buffer

The transmitter buffer includes additional circuitry to improve signal integrity, such as the programmable differential output voltage (V_{OD}), programmable three-tap pre-emphasis circuitry, internal termination circuitry, and PCIe receiver detect capability to support a PCIe configuration.

Modifying programmable values within transmitter output buffers can be performed by a single reconfiguration controller for the entire FPGA, or multiple reconfiguration controllers if desired. Within each transceiver bank a maximum of two reconfiguration controllers is allowed; one for the three-transceiver triplet in the upper-half of the bank, and one for the lower-half. This is due to a single slave interface to all PLLs and PMAs within each triplet. Therefore, many triplets can be connected to a single reconfiguration controller, but only one reconfiguration controller can be connected to the three transceivers within any triplet.

A maximum of one reconfiguration controller is allowed per transceiver bank upper-half or lower-half triplet.

The Stratix V GT transmitter buffer has only two taps for the pre-emphasis.

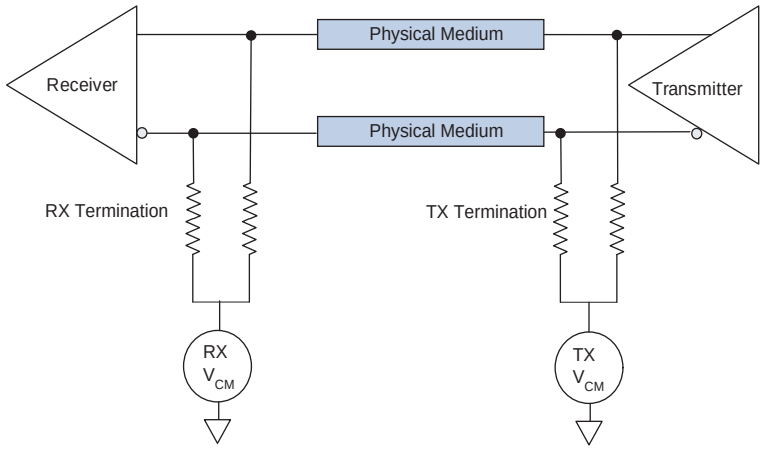
For more information, refer to the *Reconfiguration Controller to PHY IP Connectivity* section in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

Link Coupling

A high-speed serial link can be AC-coupled or DC-coupled, depending on the serial protocol you are implementing. The AC-coupled protocol is the default setting.

In a DC-coupled link, the transmitter DC V_{CM} is seen unblocked at the receiver buffer. The link V_{CM} depends on the transmitter V_{CM} and the receiver V_{CM}. The on-chip or off-chip receiver termination and biasing circuitry must ensure compatibility between the transmitter and receiver V_{CM}. Figure 1–12 shows a DC-coupled link.

Figure 1–12. DC-Coupled Link



You can DC-couple the GX channel transmitter to a GX channel receiver for the entire data rate range from 600 Mbps to 14.1 Gbps.

The GT channels only support AC coupling.

For information about transmitter and receiver common mode values, refer to the *Stratix V Device Datasheet*.

Programmable Transmitter Termination

Transmitter buffers include programmable on-chip differential termination of 85, 100, 120, 150 Ω or OFF, for GX channels The transmitter buffer of GT channels only supports 100 Ω differential termination, but provides for fine adjustment. The resistance is adjusted by the on-chip calibration circuit during calibration, which compensates for PVT changes. The transmitter buffers are current mode drivers. Therefore, the resultant V_{OD} is a function of the transmitter termination value.

You can disable OCT and use external termination. If you select external termination, the transmitter common mode is tri-stated. Common mode is based on the external termination connection.

Transmitter Analog Settings

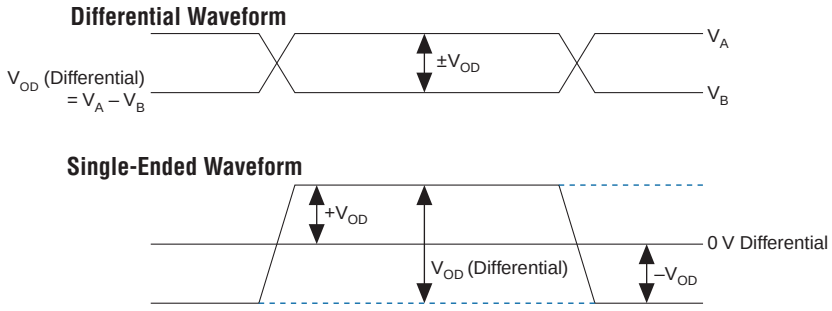
This section describes the transmitter analog setting capability to improve signal integrity.

For more information, refer to the *Altera Transceiver PHY IP Core User Guide*.

Programmable Output Differential Voltage

You can program the differential output voltage to handle different trace lengths, various backplanes, and receiver requirements. Figure 1–13 shows single-ended and differential waveforms.

Figure 1–13. V_{OD} (Differential) Signal Level



Stratix V GX channels have 64 V_{OD} settings, GT channels have 6 settings.

For information about V_{OD} values, refer to the *Stratix V Device Datasheet*.

Programmable Pre-Emphasis

Pre-emphasis can maximize the data eye opening at the far-end receiver. The programmable pre-emphasis module in each transmit buffer boosts high frequencies in the transmit data signal, to compensate attenuation in the transmission media.

Stratix V GX channels provide three pre-emphasis taps: pre-tap (16 settings), first post-tap (32 settings), and second post-tap (16 settings). The pre-tap sets the pre-emphasis on the data bit before the transition. The first post-tap and second post-tap set the pre-emphasis on the transition bit and the following bit, respectively. The pre-tap and second post-tap also provide inversion control, shown by negative values.

The Stratix V GT channels have two pre-emphasis taps: pre-tap and post-tap. Both taps have 32 settings.

Serializer

The serializer converts the incoming low-speed parallel data from the transceiver PCS to high-speed serial data and sends the data to the transmitter buffer. The Stratix V GX channel serializer supports an 8- and 10-bit, 16- and 20-bit, 32- and 40-bit, and 64-bit serialization factor. By default, the serializer block sends out the LSB of the input data first. For Stratix V GT channels, the serializer only supports a 128-bit serialization factor.

PCIe Receiver Detect

For a PCIe configuration for Gen1, Gen2, and Gen3 data rates, the transmitter buffers have a built-in receiver detection circuit. This receiver detection circuit detects if there is a receiver downstream by sending out a pulse on the common mode of the transmitter and monitoring the reflection.

PCIe Electrical Idle

The transmitter output buffers support transmission of PCIe electrical idle (or individual transmitter tri-state).

For more information about PCIe receiver detect and PCIe electrical idle, refer to the *PCI Express PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

For more information about the PCIe protocol, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

Transceiver Calibration Blocks

The following sections describe the Stratix V transceiver calibration blocks.

OCT Calibration

Stratix V GT, GX, and GS devices contain calibration circuits that calibrate the OCT resistors and analog portions of the transceiver blocks to ensure that the functionality is independent of process, voltage, and temperature (PVT) variation.

The calibration block internally generates a constant reference voltage, independent of PVT variation. The calibration block uses the internal reference voltage and external reference resistor to generate constant reference currents.

You must connect a separate 1.8 kΩ (maximum tolerance ±1%) external resistor on each RREF pin to ground. To ensure the calibration block operates properly, the RREF resistor connection on the board must be free from external noise.

Offset Cancellation in the Receiver Buffer and Receiver CDR

Process variation in smaller process silicon can lead to a V_{CM} offset between the p and n signals within the differential buffers. Stratix V devices have an automatic calibration in their receiver buffers to remove this V_{CM} offset.

You must use the reconfiguration controller IP for offset cancellation to take place. Calibration does not occur during transceiver reset, only during device configuration. Any signals that may appear on the receiver pin do not affect calibration because the receiver buffers are disconnected during calibration.

A maximum of one reconfiguration controller is allowed per transceiver bank upper-half or lower-half triplet.

ATX PLL Calibration

ATX PLL calibration optimizes the ATX PLL VCO settings for the desired output frequency. The reconfiguration controller IP must be instantiated for this calibration to run. The calibration occurs one time after device initialization.

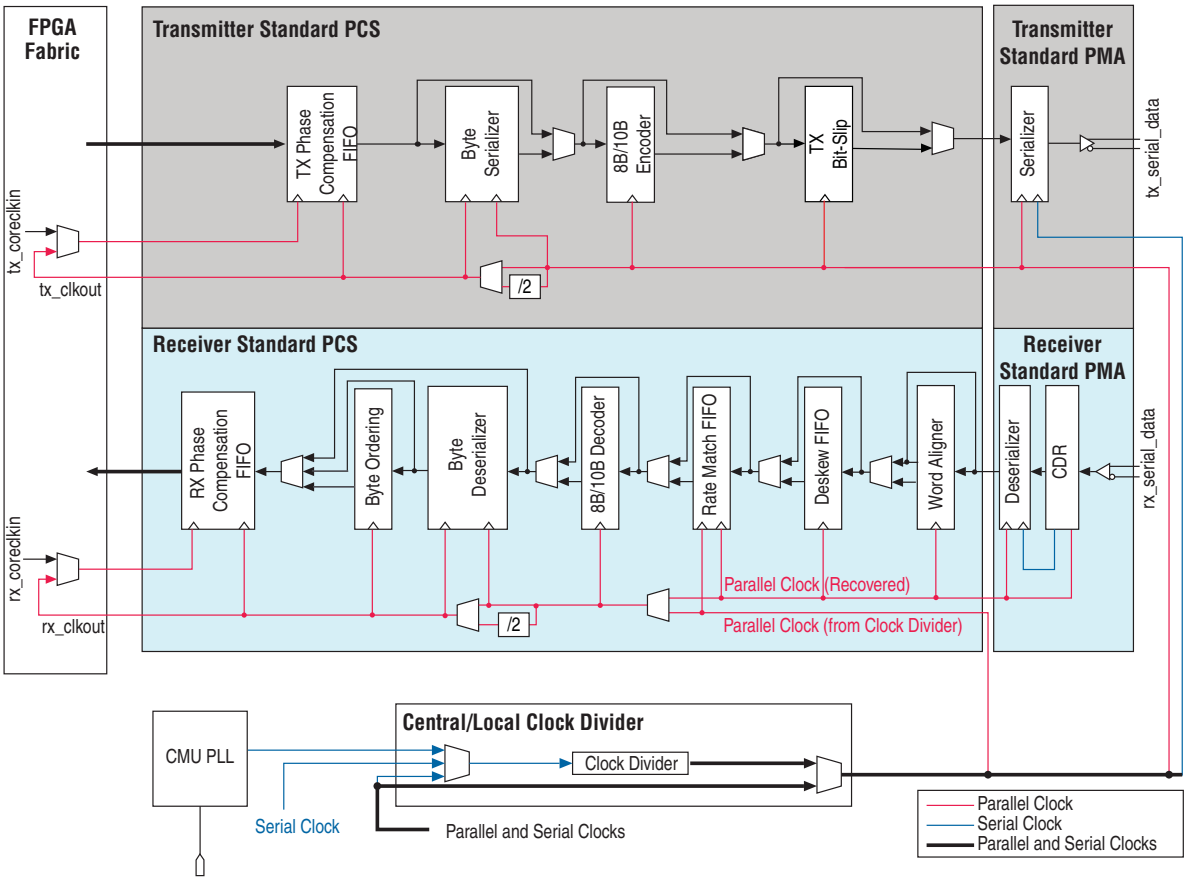
For more information, refer to the *Dynamic Reconfiguration in Stratix V Devices* chapter.

Standard PCS Architecture

This section describes the transceiver PCS circuit blocks supporting data rates up to 11 Gbps.

Figure 1–14 shows the standard PCS transceiver channel datapath, including the PMA-PCS and PCS-FPGA fabric interfaces for Stratix V GX channels.

Figure 1–14. Standard PCS Datapath in Stratix V GX Channels



The PCS can be completely bypassed using the PMA direct mode. This is an option for GX channels. The GT channels do not have a PCS, and operates in PMA direct mode.

The following sections describe the standard PCS architecture on GX channels:

- "Receiver Standard PCS Datapath" on page 1–21
- "Transmitter Standard PCS Datapath" on page 1–30

Some transceiver channels interface to the PCIe hard IP block, the PIPE interface for soft IP implementations of PCIe, or directly to the FPGA fabric (FPGA fabric-transceiver interface). The transceiver channel interfaces to the PCIe hard IP block if you use the hard IP block to implement the PCIe PHY MAC, data link layer, and transaction layer. Otherwise, the transceiver channel interfaces directly to the FPGA fabric.

The PCIe hard IP-transceiver interface is outside the scope of this chapter. This chapter describes the FPGA fabric-transceiver interface only.

For more information about the PCIe hard IP block, refer to the "PCI Express PHY IP Core" chapter in the *Altera Transceiver PHY IP Core User Guide* and to the *IP Complier for PCI Express Compiler User Guide*.

You can divide the standard transceiver channel datapath into two configurations based on the FPGA fabric-transceiver interface width (channel width) and the transceiver channel PMA-PCS width (serialization factor):

Table 1–1 lists the FPGA fabric-transceiver interface widths (channel width) and transceiver PMA-PCS widths (serialization factor) allowed.

Table 1–1. Transceiver Interface Widths for Stratix V Devices

Name	8- and 10-Bit PMA-PCS Widths	16- and 20-Bit PMA-PCS Widths
FPGA fabric-transceiver interface widths	8 and 10 bit 16 and 20 bit	16 and 20 bit 32 and 40 bit
Supported configurations	PCIe Gen1, Gen2, and Gen3 XAUI Custom	Custom
Data rate range in a custom configuration	0.6 to 3.75 Gbps	1.0 to 11 Gbps

Receiver Standard PCS Datapath

This section describes the receiver channel PCS datapath architecture. The functional blocks in the receiver datapath are described in order from the word aligner to the receiver phase compensation FIFO buffer at the FPGA fabric-transceiver interface. Figure 1–14 on page 1–20 shows the receiver GX channel datapath.

Table 1–2 lists the receiver channel components.

Table 1–2. Receiver Channel Components for Stratix V Devices (Part 1 of 2)

Component	Refer To:
Word Aligner	"Word Aligner" on page 1–22
Deskew FIFO	"Deskew FIFO" on page 1–25
Rate Match (Clock Rate Compensation) FIFO	"Rate Match (Clock Rate Compensation) FIFO" on page 1–26

Table 1–2. Receiver Channel Components for Stratix V Devices (Part 2 of 2)

Component	Refer To:
8B/10B Decoder (for PCIe configurations only)	"8B/10B Decoder" on page 1–26
Byte Deserializer	"Byte Deserializer" on page 1–27
Byte Ordering Block	"Byte Ordering Block" on page 1–28
Receiver Phase Compensation FIFO	"Receiver Phase Compensation FIFO" on page 1–30

The receiver datapath is flexible and allows multiple modes, depending on the selected configuration.

Word Aligner

Because the data is serialized before transmission and then deserialized at the receiver, the data loses the word boundary of the upstream transmitter after deserialization. The word aligner receives parallel data from the deserializer and restores the word boundary based on a pre-defined alignment pattern that must be received during link synchronization.

Serial protocols such as PCIe specify a standard word alignment pattern. For proprietary protocols, the transceiver architecture allows you to select a custom word alignment pattern specific to your implementation.

In addition to restoring the word boundary, the word aligner also implements the following features:

- Synchronization state machine in configurations such as PCIe
- Programmable run length violation detection in all configurations
- Receiver polarity inversion in all configurations except PCIe
- Receiver bit reversal in custom configurations
- Receiver byte reversal in custom 16- and 20-bit width configurations

Depending on the configuration, the word aligner operates in one of the following three modes:

- Manual alignment
- Automatic synchronization state machine
- Bit-slip

Table 1–3 lists the available word aligner options.

Table 1–3. Word Aligner Options Available for Stratix V Devices

Configuration	PMA-PCS Interface Width (Bits)	Word Alignment Mode	Word Alignment Pattern Length	Word Alignment Behavior
Custom 8- and 10-Bit Width	8	Manual Alignment	16 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is reasserted.
		Bit-Slip	16 bits	User-controlled signal shifts data one bit at a time.
	10	Manual Alignment	7 and 10 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is reasserted.
		Bit-Slip	7 and 10 bits	User-controlled signal shifts data one bit at a time.
		Automatic Synchronized State Machine	7 and 10 bits	Data is required to be 8B/10B encoded. Aligns to the selected word aligner pattern.
Custom 16- and 20-Bit Width	16	Manual Alignment	8, 16, and 32 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is reasserted.
		Bit-Slip	8, 16, and 32 bits	User-controlled signal shifts data one bit at a time.
	20	Manual Alignment	7, 10, and 20 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is reasserted.
		Bit-Slip	7, 10, and 20 bits	User-controlled signal shifts data one bit at a time.
		Automatic Synchronized State Machine	7 and 10 bits	Data is required to be 8B/10B encoded. Aligns to the selected word aligner pattern.
PCIe	10	Automatic Synchronized State Machine	10 bits	Automatically selected word aligner pattern length and pattern.

Bit-Slip Mode Word Aligner with an 8-Bit PMA-PCS Interface Configuration

In a custom width configuration with an 8-bit PMA-PCS interface width, you can configure the word aligner in bit-slip mode. In bit-slip mode, the word aligner operation is controlled by the `rx_bitslip` bit of the `pcs8g_rx_wa_control` register. At every 0-1 transition of the `rx_bitslip` bit of the `pcs8g_rx_control` register, the bit-slip circuitry slips one bit into the received data stream, effectively shifting the word boundary by one bit. Also in bit-slip mode, the word aligner `pcs8g_rx_wa_status` register bit for `rx_patterndetect` is driven high for one parallel clock cycle when the received data after bit-slipping matches the 16-bit word alignment pattern programmed.

You can implement a bit-slip controller in the FPGA fabric that monitors the `rx_parallel_data` signal, the `rx_patterndetect` signal, or both, and controls the `rx_bitslip` signal to achieve word alignment.

Automatic Synchronization State Machine Mode Word Aligner with a 10-Bit PMA-PCS Interface Configuration

Protocols such as PCIe require the receiver PCS logic to implement a synchronization state machine to provide hysteresis during link synchronization. Each of these protocols defines a specific number of synchronization code groups that the link must receive to acquire synchronization, and a specific number of erroneous code groups that the protocol must receive to fall out of synchronization.

In PCIe configurations, the word aligner is in automatic synchronization state machine mode. The word aligner automatically selects the word alignment pattern length and pattern as specified by each protocol.

Table 1–4 lists the synchronization state machine modes. The synchronization state machine parameters are fixed for PCIe configurations as specified by the respective protocol.

Table 1–4. Synchronization State Machine Modes for Stratix V Devices

Mode	PCIe
Number of valid synchronization code groups or ordered sets received to achieve synchronization	4
Number of erroneous code groups received to lose synchronization	17
Number of continuous good code groups received to reduce the error count by one	16

After deassertion of the `reset_rx_digital` signal in automatic synchronization state machine mode, the word aligner starts looking for the word alignment pattern or synchronization code groups in the received data stream. When the programmed number of valid synchronization code groups or ordered sets is received, the `rx_syncstatus` status bit is driven high to indicate that synchronization is acquired. The `rx_syncstatus` status bit is constantly driven high until the programmed number of erroneous code groups is received without receiving intermediate good groups; after which `rx_syncstatus` is driven low. The word aligner indicates loss of synchronization (`rx_syncstatus` remains low) until the programmed number of valid synchronization code groups are received again.

Programmable Run Length Violation Detection

The programmable run length violation circuit resides in the word aligner block and detects consecutive 1s or 0s in the data. If the data stream exceeds the preset maximum number of consecutive 1s or 0s, the violation is signified by the assertion of the `rx_rlv` status bit.

Table 1–5 lists the detection capabilities of the run length violation circuit.

Table 1–5. Detection Capabilities of the Run Length Violation Circuit for Stratix V Devices

PMA-PCS Interface Width (Bits)	Run Length Violation Detector Range	
	Minimum	Maximum
8	4	128
10	5	160
16	8	512
20	10	640

Receiver Polarity Inversion

The positive and negative signals of a serial differential link may be erroneously swapped during board layout. Solutions such as board re-spin or major updates to the PLD logic are expensive. The receiver polarity inversion feature is provided to correct this situation.

Receiver Bit Reversal

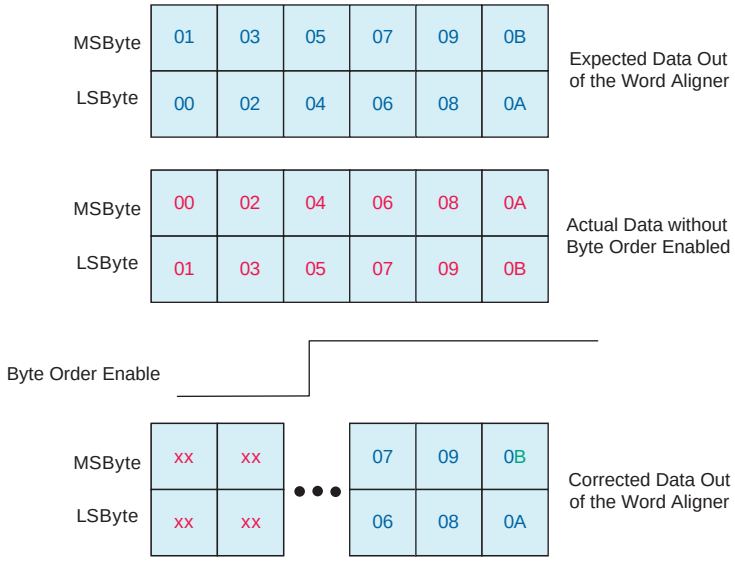
By default, the receiver assumes an LSB-to-MSB transmission. If the transmission order is MSB-to-LSB, the receiver forwards the bit-flipped version of the parallel data to the FPGA fabric on `rx_parallel_data`. For example, if in 8 bit width mode, `D[7:0]` is rewired to `D[0:7]`.

Receiver Byte Reversal in Custom 16- and 20-Bit Width Configurations

The MSByte and LSByte of the input data to the transmitter may be erroneously swapped. The receiver byte reversal feature corrects this situation.

Figure 1–15 shows the receiver byte reversal feature.

Figure 1–15. Receiver Byte Reversal Feature



Deskew FIFO

The deskew FIFO in each channel receives data from its word aligner. The deskew FIFO circuitry aligns the data across multiple channels.

The deskew circuitry is available only in XAUI mode.

In the XAUI protocol, the code groups received across four lanes of a XAUI link can be misaligned with respect to one another because of skew in the physical medium or differences between the independent clock recoveries per lane.

The XAUI protocol requires the physical layer device to implement a deskew circuitry to align all four channels. To enable the deskew circuitry at the receiver, the transmitter sends a /A/ (/K28.3/) code group simultaneously on all four channels during inter-packet gap (IPG). The skew introduced in the physical medium and the receiver channels can cause the /A/ code groups to be misaligned when received.

The deskew operation begins only after link synchronization is achieved on all four channels from the word aligner of each channel. When an aligned set of /A/ code groups is observed at the output of the deskew FIFOs of the four channels, the `rx_channelaligned` signal is asserted high, indicating channel alignment is acquired.

Rate Match (Clock Rate Compensation) FIFO

The rate match (clock rate compensation) FIFO compensates for small clock frequency differences between the upstream transmitter and the local receiver clocks by inserting or removing skip (SKP) symbols or ordered sets from the inter-packet gap (IPG) or idle streams. The rate match FIFO deletes SKP symbols or ordered sets when the upstream transmitter reference clock frequency is higher than the local receiver reference clock frequency. The rate match FIFO inserts SKP symbols or ordered sets when the local receiver reference clock frequency is higher than the upstream transmitter reference clock frequency.

For more information about how to use the rate match FIFO with PCIe, XAUI, and Custom protocols, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

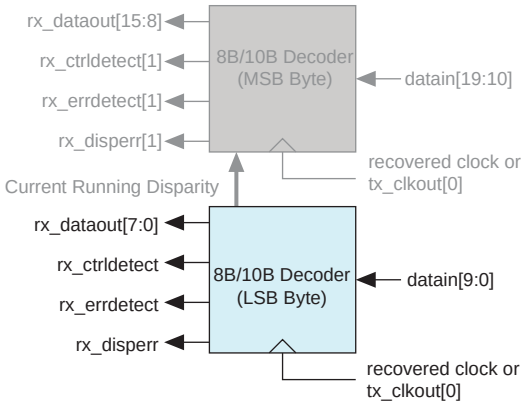
8B/10B Decoder

PCIe (Gen 1 and Gen2) and XAUI modes require the serial data sent over the link to be 8B/10B encoded to maintain the DC balance in the transmitted serial data. The PCIe protocol requires the receiver PCS logic to implement an 8B/10B decoder to decode the data before forwarding the data to the upper layers for packet processing.

The receiver channel PCS has an 8B/10B decoder after the rate match FIFO. In configurations with the rate match FIFO enabled, the 8B/10B decoder receives data from the rate match FIFO. In configurations with the rate match FIFO disabled, the 8B/10B decoder receives data from the word aligner.

Figure 1–16 shows the 8B/10B decoder in 10-bit mode. In 10-bit mode, the 8B/10B decoder receives 10-bit data from the rate match FIFO or word aligner (when the rate match FIFO is disabled) and decodes the data into an 8-bit data +1-bit control identifier. The decoded data is fed to the byte deserializer or the receiver phase compensation FIFO (if the byte deserializer is disabled).

Figure 1–16. 8B/10B Decoder in 10-bit Mode



The 8B/10B decoder is described in IEEE 802.3-2008 clause-49.

The 8B/10B decoder operates in 10-bit width mode only in PCIe configuration. A PCIe configuration forces selection of the 8B/10B decoder in the receiver datapath.

Control Code Group Detection

The 8B/10B decoder indicates whether the decoded 8-bit code group is a data or control code group on the `rx_datak` signal (not shown in Figure 1–16). If the received 10-bit code group is one of the 12 control code groups (/Kx.y/) specified in the IEEE 802.3 specification, the `rx_datak` signal is driven high. If the received 10-bit code group is a data code group (/Dx.y/), the `rx_datak` signal is driven low.

Byte Deserializer

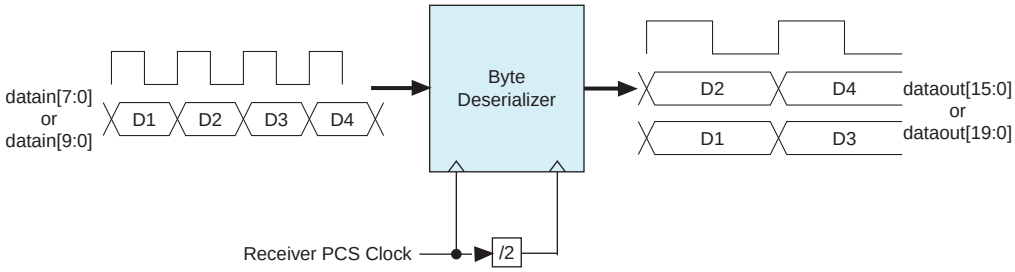
The FPGA fabric-transceiver interface frequency has an upper limit. In designs where the receiver PCS frequency exceeds the upper limit, the byte deserializer is required. The byte deserializer reduces the interface frequency to half while doubling the parallel data width. The byte deserializer is optional in designs that do not exceed the interface frequency upper limit.

Byte Deserializer in 8- and 10-Bit Width Mode

In 8-bit width mode, the byte deserializer receives 8-bit wide data from the 8B/10B decoder or 10-bit wide data from the word aligner (if the 8B/10B decoder is disabled) and deserializes the data into 16- or 20-bit wide data at half the speed.

Figure 1–17 shows the byte deserializer in 8- or 10-bit width mode.

Figure 1–17. Byte Deserializer in 8- or 10-Bit Width Mode

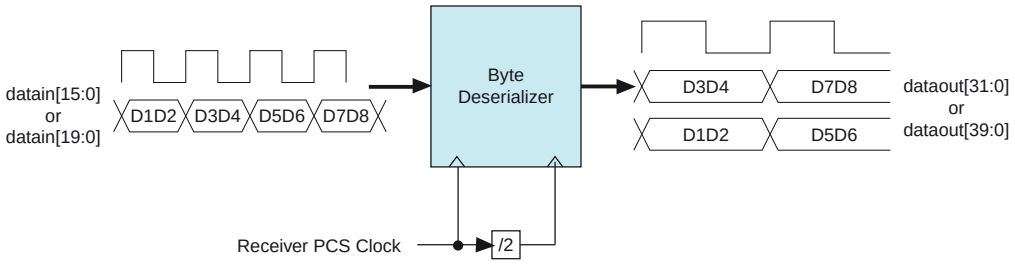


Byte Deserializer in 16- or 20-Bit Width Mode

In 16-bit width mode, the byte deserializer receives 16-bit wide data from the 8B/10B decoder or 20-bit wide data from the word aligner (if the 8B/10B decoder is disabled) and deserializes the data into 32- or 40-bit wide data at half the speed.

Figure 1–18 shows the byte deserializer in 16- or 20-bit width mode.

Figure 1–18. Byte Deserializer in 16- and 20-Bit Width Mode



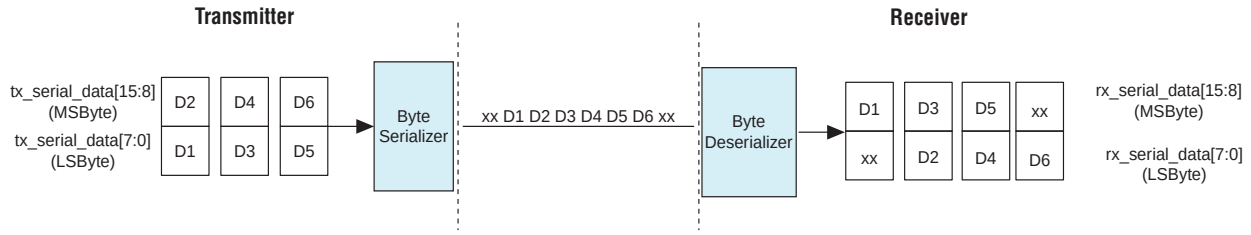
Byte Ordering Block

In 8- or 10-bit width mode with the 16- or 20-bit FPGA fabric-transceiver interface, the byte deserializer receives one data byte (8 or 10 bits) and deserializes the data into two data bytes (16 or 20 bits). Depending on when the receiver PCS logic comes out of reset, the byte ordering at the output of the byte deserializer may or may not match the original byte ordering of the transmitted data. The byte misalignment resulting from byte deserialization is unpredictable because the byte misalignment depends on which byte is being received by the byte deserializer when the byte comes out of reset.

The byte ordering block looks for the user-programmed byte ordering pattern in the byte-deserialized data. You must select a byte ordering pattern that you know is at the LSBytes position of the parallel transmitter data. If the byte ordering block finds the programmed byte ordering pattern in the MSBytes position of the byte-deserialized data, the byte ordering block inserts the appropriate number of user-programmed pad bytes to push the byte ordering pattern to the LSByte(s) position, thereby restoring proper byte ordering.

Figure 1–19 shows a scenario in which the MSByte and LSByte of the two-byte transmitter data appears straddled across two word boundaries after being byte deserialized at the receiver.

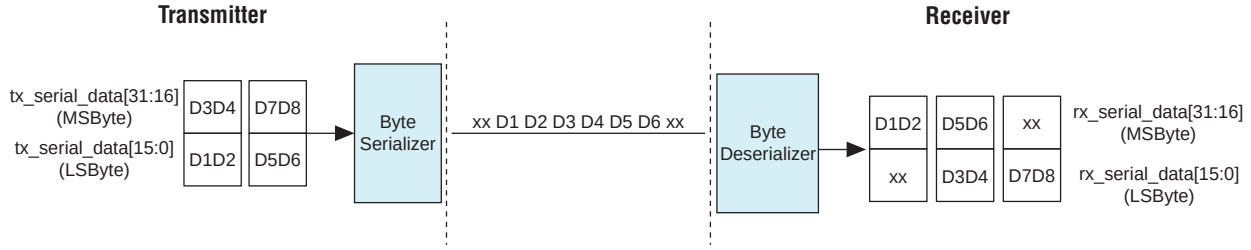
Figure 1–19. MSByte and LSByte of the Two-Bit Transmitter Data Straddled Across Two Word Boundaries



In 16-bit width mode with a 32-bit FPGA fabric-transceiver interface, the byte deserializer receives two data bytes (16 bits) and deserializes the two data bytes into four data bytes (32 bits).

Figure 1–20 shows a scenario in which the two MSBytes and LSBytes of the four-byte transmitter data appears straddled across two word boundaries after being byte deserialized at the receiver.

Figure 1–20. MSByte and LSByte of the Four-Bit Transmitter Data Straddled Across Two Word Boundaries



Byte Ordering Block in 8- or 10-Bit Width Modes

In a custom 8- or 10-bit width configuration, you can program a custom byte ordering pattern and byte ordering pad pattern in the PHY IP megafunction. Table 1–6 lists the byte ordering pattern length allowed in a custom 8-bit width configuration.

Table 1–6. Byte Ordering Pattern Length in a Custom 8-Bit Width Configuration for Stratix V Devices

Configuration	Byte Ordering Pattern Length (Bits)	Byte Ordering Pad Pattern Length (Bits)
Custom 8-bit configuration with: ■ 16-bit FPGA fabric-transceiver interface ■ No 8B/10B decoder ■ Word aligner in manual alignment mode	8	8

Byte Ordering Block in 16- or 20-Bit Width Modes

In a custom 16- or 20-bit width configuration, you can program a custom byte ordering pattern and byte ordering pad pattern in the PHY IP megafunction. Table 1–7 lists the byte ordering pattern length allowed in a custom 16- or 20-bit width configuration.

Table 1–7. Byte Ordering Pattern Length in a Custom 16-Bit Width Configuration for Stratix V Devices

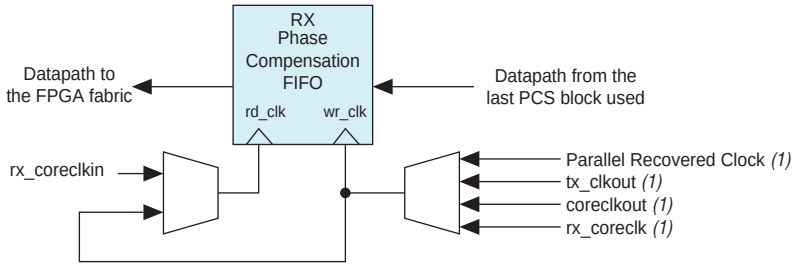
Configuration	Byte Ordering Pattern Length (Bits)	Byte Ordering Pad Pattern Length (Bits)
Custom 16-bit width configuration with: ■ 32-bit FPGA fabric-transceiver interface ■ No 8B/10B decoder (16-bit PMA-PCS interface) ■ Word aligner in manual alignment mode	16 and 8	8

Receiver Phase Compensation FIFO

The receiver phase compensation FIFO in each channel ensures the reliable transfer of data and status signals between the receiver channel and the FPGA fabric. The receiver phase compensation FIFO compensates for the phase difference between the parallel receiver PCS clock (FIFO write clock) and the FPGA fabric clock (FIFO read clock).

Figure 1–21 shows the receiver phase compensation FIFO.

Figure 1–21. Receiver Phase Compensation FIFO



Note to Figure 1–21:

(1) If you use the byte deserializer, these clocks are divided by two.

Transmitter Standard PCS Datapath

This section describes the transmitter standard PCS datapath architecture. Table 1–8 lists the transmitter channel components.

Table 1–8. Transmitter Channel Components for Stratix V Devices

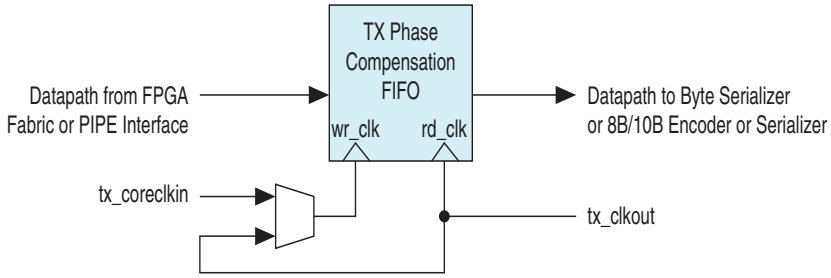
Component	Refer To:
Transmitter Phase Compensation FIFO	"Transmitter Phase Compensation FIFO"
Byte Serializer	"Byte Serializer" on page 1–31
8B/10B Encoder	"8B/10B Encoder" on page 1–32

The Standard PCS is not supported in the GT channels.

Transmitter Phase Compensation FIFO

The transmitter phase compensation FIFO interfaces with the FPGA fabric or PCIe interface. The transmitter phase compensation FIFO compensates for the phase difference between the low-speed parallel clock and the FPGA fabric interface clock. Figure 1–22 shows the datapath and clocking of the transmitter phase compensation FIFO.

Figure 1–22. Transmitter Phase Compensation FIFO



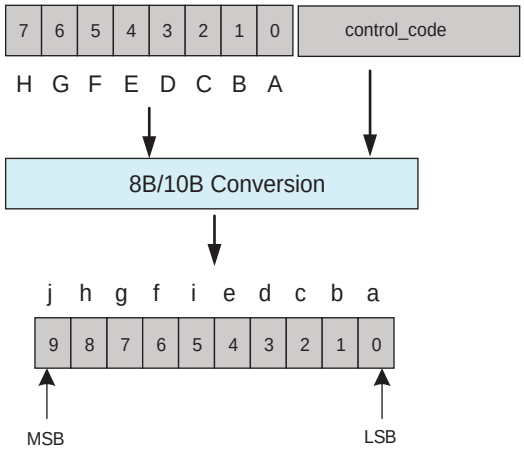
Byte Serializer

The byte serializer allows you to run the transceiver channel at higher data rates while keeping the FPGA fabric interface frequency below the upper limit, by halving the width of the data bus and doubling the data rate. The Byte Serializer forwards the least significant word first followed by the most significant word. For example, assuming a channel width of 32, the byte serializer forwards `tx_parallel_data[15:0]` first, followed by `tx_parallel_data[31:16]`.

8B/10B Encoder

The 8B/10B encoder generates 10-bit code groups from the 8-bit data and 1-bit control identifier. In 8-bit width mode, the 8B/10B encoder translates the 8-bit data to a 10-bit code group (control word or data word) with proper disparity. If the `tx_datak` input is high, the 8B/10B encoder translates the input `data[7:0]` to a 10-bit control word. If the `tx_datak` input is low, the 8B/10B encoder translates the input `data[7:0]` to a 10-bit data word. Figure 1–23 shows the conversion format. The LSB is transmitted first.

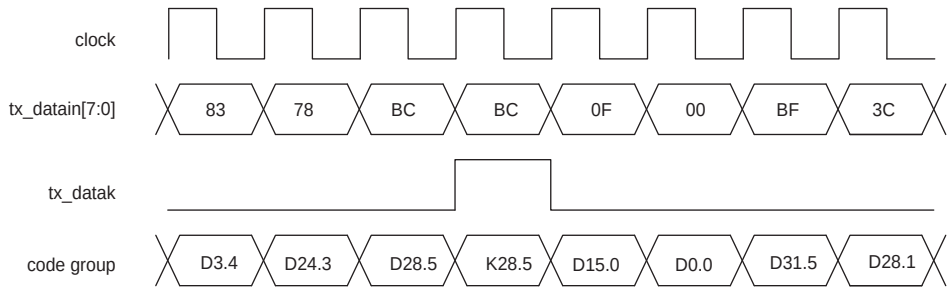
Figure 1–23. 8B/10B Conversion Format



Control Code Encoding

The IEEE 802.3 8B/10B encoder specification identifies only a set of 8-bit characters for which `tx_datak` must be asserted. If you assert `tx_datak` for any other set of bytes, the 8B/10B encoder might encode the output 10-bit code as an invalid code (it does not map to a valid Dx.y or Kx.y code), or unintended valid Dx.y code, depending on the value entered. It is possible for a downstream 8B/10B decoder to decode an invalid control word into a valid Dx.y code without asserting code error flags. Figure 1–24 shows the transmission of both the control and data words.

Figure 1–24. Control Word and Data Word Transmission



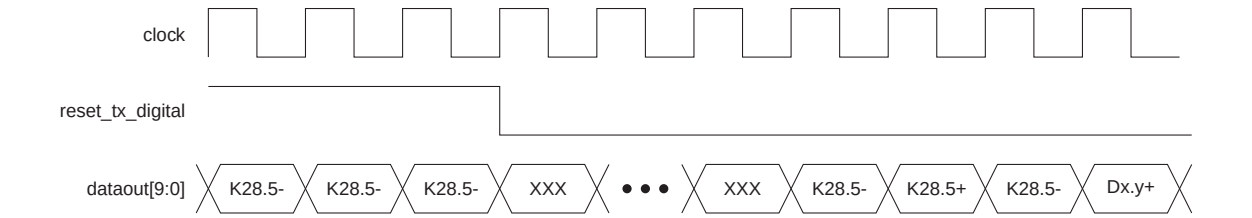
Reset Condition

The reset_tx_digital signal resets the 8B/10B encoder. During reset, running disparity and data registers are cleared. Also, the 8B/10B encoder outputs a K28.5 pattern from the RD- column continuously until reset_tx_digital is deasserted. The input data and control code from the FPGA fabric is ignored during the reset state. After reset, the 8B/10B encoder starts with a negative disparity (RD-) and transmits three K28.5 code groups for synchronization before it starts encoding and transmitting the data on its output.

While reset_tx_digital is asserted, the downstream 8B/10B decoder that receives the data may observe synchronization or disparity errors.

Figure 1–25 shows the reset behavior of the 8B/10B encoder. When in reset (reset_tx_digital is high), a K28.5- (K28.5 10-bit code group from the RD-column) is sent continuously until reset_tx_digital is low. Because of some pipelining of the transmitter channel PCS, some "don't cares" (10'hxxx) are sent before the three synchronizing K28.5 code groups. User data follows the third K28.5 code group.

Figure 1–25. 8B/10B Encoder Output During reset_tx_digital Deassertion



Transmitter Polarity Inversion

The positive and negative signals of a serial differential link may be erroneously swapped during board layout. Solutions such as board re-spin or major updates to the PLD logic are expensive. The transmitter polarity inversion feature of the 8B/10B encoder is provided to correct this situation.

Transmitter Bit-Slip

The transmitter bit-slip allows you to compensate for the channel-to-channel skew between multiple transmitter channels by slipping the data sent to the PMA.

10G PCS Architecture

The 10G PCS architecture offers a full duplex (transmitter and receiver) transceiver channel that supports serial data rates up to 14.1 Gbps for Stratix V GX and GS devices and up to 12.5 Gbps for Stratix V GT devices.

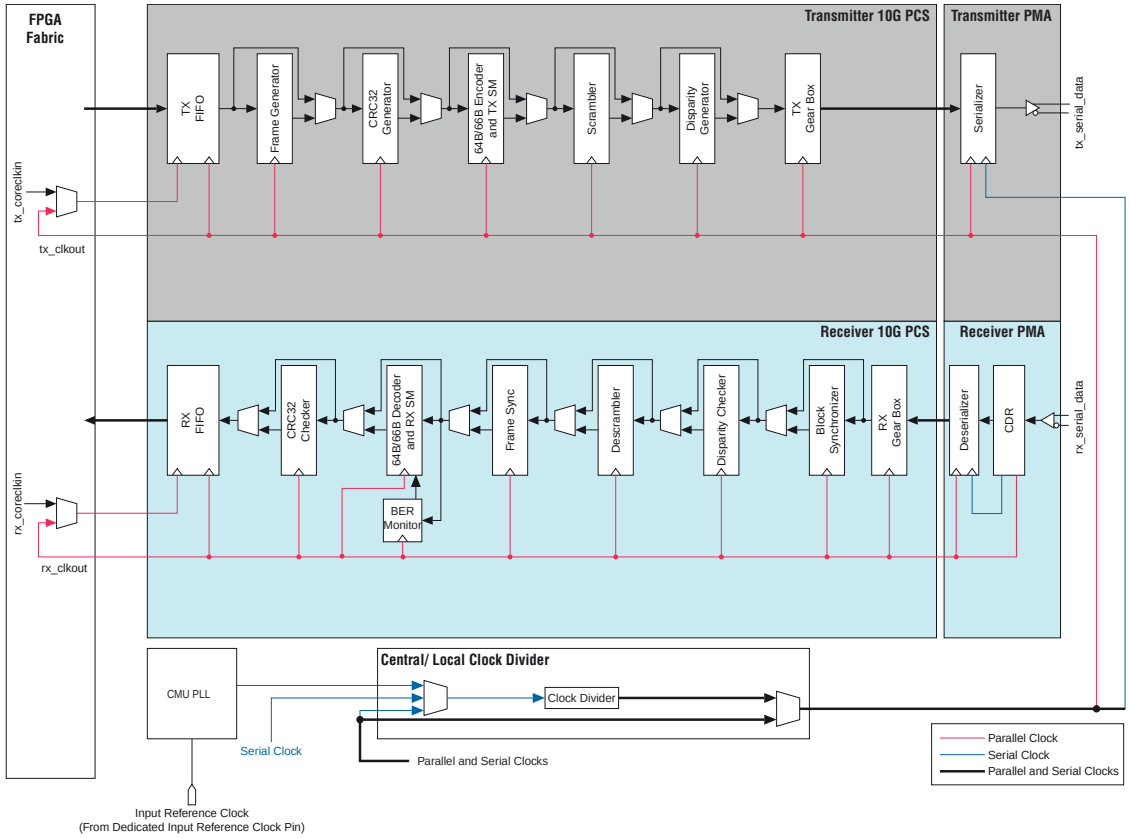
Several functional blocks are customized for various protocols. The different datapath configurations for these protocols are available through the different PHY IPs instantiated through the MegaWizard™ Plug-In Managers. Currently, the only supported configurations for the 10G PCS are Interlaken, 10G-BASE-R and low-latency 10G PCS.

For more information about 10G PCS configurations, and clocking schemes and placement restrictions for the different datapath configurations supported in Stratix V devices, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

The functional blocks in the 10G PCS hard macro provide status and control signals to the FPGA fabric. For signal names, refer to the *Altera Transceiver PHY IP Core User Guide*.

Figure 1–26 shows the 10G PCS datapath for Stratix V GX channels.

Figure 1–26. 10G PCS Datapath in Stratix V GX Channels *(1)*



Note to Figure 1–26:

(1) Not all the blocks shown in the 10G PCS datapath are available in every configuration.

The following sections describe the 10G PCS architecture:

- "Receiver 10G PCS Datapath" on page 1–35
- "Transmitter 10G PCS Datapath" on page 1–38

Receiver 10G PCS Datapath

This section describes the receiver 10G PCS datapath architecture. The sub-blocks in the receiver 10G PCS datapath are described in order from the receiver gearbox to the receiver FIFO. Table 1–9 lists the receiver 10G PCS channel components.

Table 1–9. Receiver 10G PCS Channel Components for Stratix V Devices

Component	Refer To:
Receiver Gearbox	"Receiver Gearbox" on page 1–35
Block Synchronizer	"Block Synchronizer" on page 1–35
Disparity Checker	"Disparity Checker" on page 1–36
Descrambler	"Descrambler" on page 1–36
Frame Synchronizer	"Frame Synchronizer" on page 1–36
Bit-Error Rate (BER) Monitor	"Bit-Error Rate (BER) Monitor" on page 1–36
64B/66B Decoder	"64B/66B Decoder" on page 1–37
CRC-32 Checker	"CRC-32 Generator" on page 1–39
Receiver FIFO	"Receiver FIFO" on page 1–37

Receiver Gearbox

The receiver gearbox adapts the PMA data width to a larger bus width to interface with the PCS.

The PMA bus width is smaller than the PCS bus width; therefore, the receiver gearbox expands the data bus width from the PMA to the PCS. Because bus width adaptation is transparent, you can continuously feed data to the receiver gearbox. In addition to providing bus width adaptation, the receiver gearbox provides the receiver bit reversal feature.

Receiver Polarity Inversion

The receiver gearbox can invert the polarity of the incoming data. This is useful if the receive signals are reversed on the board or backplane layout.

Receiver Bit Reversal

The receiver gearbox allows bit reversal of the received data. Some protocols, such as Interlaken, require the bit reversal feature. For more information, refer to "Transmitter Bit Reversal" on page 1–42.

Block Synchronizer

The block synchronizer determines the block boundary of a 66-bit word in the case of the 10GBASE-R protocol or a 67-bit word in the case of the Interlaken protocol. The incoming data stream is slipped one bit at a time until a valid synchronization header (bits 65 and 66) is detected in the received data stream. After the predefined number of synchronization headers (as required by the protocol specification) is detected, the block synchronizer asserts the status signal to other receiver PCS blocks down the receiver datapath and to the FPGA fabric.

The block synchronizer is designed in accordance with both the Interlaken protocol specification and the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49.

Disparity Checker

The disparity checker is only used in Interlaken configurations

The design of the disparity checker is based on the Interlaken protocol specifications. After word synchronization is achieved, the disparity checker monitors the status of the 67th bit of the incoming word and determines whether or not to invert bits [63:0] of the received word.

Table 1–10 interprets the MSB in the 67-bit word.

Table 1–10. Interpretation of the MSB in the 67-Bit Payload for Stratix V Devices

MSB	Interpretation
0	Bits [63:0] are not inverted; the disparity checker processes the word without modification
1	Bits [63:0] are inverted; the disparity checker inverts the word to achieve the original word before processing it

Descrambler

The descrambler descrambles data per the protocol specifications supported by the 10G PCS. The descrambler operates either in frame synchronous or self synchronous mode.

Frame Synchronous Mode

Frame synchronous mode is used in Interlaken configurations only. When block synchronization is achieved, the descrambler uses the scrambler seed from the received scrambler state word. This block also forwards the current descrambler state to the frame synchronizer.

Self Synchronous Mode

Self synchronous mode is used in 10GBASE-R configurations only.

Frame Synchronizer

The frame synchronizer block achieves lock by looking for four synchronization words in consecutive metaframes. After synchronization, the frame synchronizer monitors the scrambler word in the metaframe and deasserts the lock signal after three consecutive mismatches and starts the synchronization process again. Lock status is available to the FPGA fabric.

The frame synchronizer is only used in Interlaken configurations.

Bit-Error Rate (BER) Monitor

The BER monitor block conforms to the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49. After block lock is achieved, the BER monitor starts to count the number of invalid synchronization headers within a 125-µs period. If more than 16 invalid synchronization headers are observed in a 125-µs period, the BER monitor provides the status signal to the FPGA fabric, indicating a high bit error rate condition.

64B/66B Decoder

The 64B/66B decoder block contains a 64B/66B decoder sub-block and a receiver state machine sub-block. The 64B/66B decoder sub-block converts the received data from the descrambler into 64-bit data and 8-bit control characters. The receiver state machine sub-block monitors the status signal from the BER monitor. If the status signal is asserted, the receiver state machine sends local fault ordered sets to the FPGA interface.

The 64B/66B encoder is used only in 10GBASE-R configurations.

The 64B/66B decoder block is designed towards the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49.

CRC-32 Checker

The CRC-32 checker block supports the Interlaken protocol. The CRC-32 checker calculates the CRC from the incoming data and compares the result to the CRC value sent in the diagnostic word. The CRC error signal is sent to the FPGA fabric.

Receiver FIFO

The receiver FIFO block operates in different modes based on the transceiver datapath configuration.

For more information about the different receiver FIFO operating modes, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

The Quartus II software automatically selects receiver FIFO mode for the configuration you use.

Clock Compensation Mode

The receiver FIFO is configured in clock compensation mode for the 10GBASE-R configuration. In clock compensation mode, the FIFO deletes idles or ordered sets and inserts only idles to compensate up to a ±100 ppm clock difference between the remote transmitter and the local receiver.

Generic Mode

The receiver FIFO is configured in generic mode for the Interlaken configuration. In generic mode, the receiver FIFO provides the FIFO partially empty and FIFO full status signals to the FPGA fabric to control the read side of the FIFO.

Phase Compensation Mode

The receiver FIFO is configured in phase compensation mode for the 10G custom configuration. In phase compensation mode, the FIFO compensates for the phase difference between the FIFO write clock and the read clock.

XAUI Mode

In XAUI mode, the receiver FIFO compensates for up to ±100 ppm (200 ppm total) difference between the upstream transmitter and the local receiver reference clock. The XAUI protocol requires the transmitter to send /R/ (/K28.0/) code groups simultaneously on all four lanes (denoted as ||R|| column) during inter-packet gaps (IPGs), conforming to the IEEE P802.3ae specification. The receiver FIFO operation in XAUI mode is compliant with the IEEE P802.3ae specification.

PCIe Mode

In PCIe mode, the receiver FIFO compensates for up to ±300 ppm (total 600 ppm) difference between the upstream transmitter and the local receiver. The PCIe protocol requires the transmitter to send SKP ordered sets during IPGs, conforming to the PCIe base specification 2.0. The SKP ordered set is defined as a /K28.5/ COM symbol followed by three consecutive /K28.0/ SKP symbol groups.

The PCIe protocol requires the receiver to recognize a SKP ordered set as a /K28.5/ COM symbol followed by one to five consecutive /K28.0/ SKP symbols. The rate match FIFO operation is compliant with PCIe Base Specification 2.0.

Transmitter 10G PCS Datapath

This section describes the transmitter 10G PCS datapath architecture. The sub-blocks in the transmitter 10G PCS datapath are described in order from the transmitter FIFO to the transmitter gearbox. Table 1–11 lists the transmitter 10G PCS channel components.

Table 1–11. Transmitter 10G PCS Channel Components for Stratix V Devices

Component	Refer To:
Transmitter FIFO	"Transmitter FIFO" on page 1–38
Frame Generator	"Frame Generator" on page 1–39
CRC-32 Generator	"CRC-32 Generator" on page 1–39
64B/66B Encoder	"64B/66B Encoder" on page 1–40
Scrambler	"Scrambler" on page 1–40
Disparity Generator	"Disparity Generator" on page 1–40
Transmitter Gearbox	"Transmitter Gearbox" on page 1–41

Transmitter FIFO

The transmitter FIFO provides an interface between the transmitter channel PCS and the FPGA fabric.

In 10GBASE-R configurations, the transmitter FIFO receives data from the FPGA fabric. The data output from the transmitter FIFO block goes to the 64B/66B encoder.

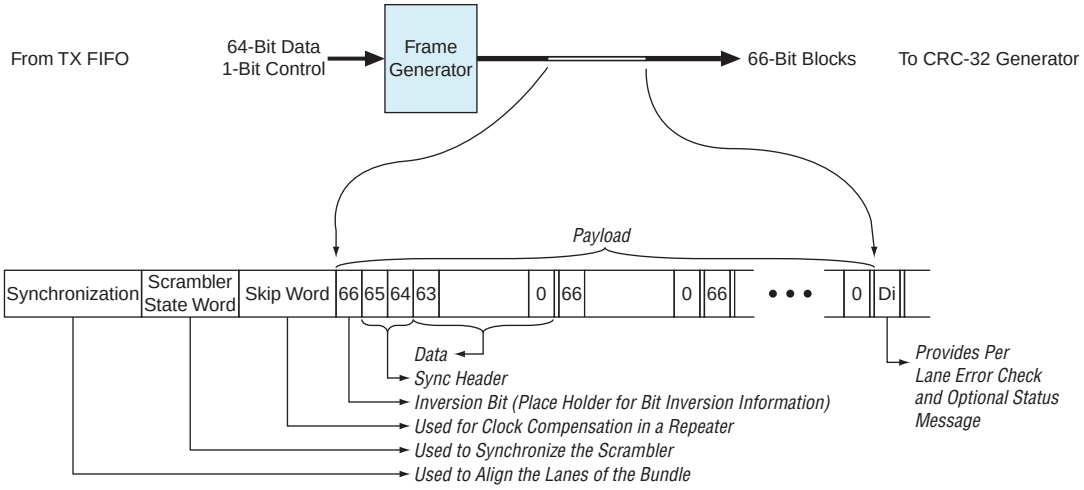
In Interlaken configurations, the transmitter FIFO sends a control signal to indicate whether it is ready to receive data from the FPGA fabric. The user logic sends the data to the transmitter FIFO only if this control signal is asserted. In this configuration, data output from the transmitter FIFO block goes to the frame generator.

Frame Generator

The frame generator is used only in Interlaken configurations.

The frame generator block (Figure 1–27) supports the Interlaken protocol. The frame generator block takes the data from the transmitter FIFO and encapsulates the payload and burst/idle control words from the FPGA fabric with the framing layer's control words, such as the synchronization word, scrambler state word, skip word, and diagnostic word, to form a metaframe. The Interlaken PHY IP MegaWizard Plug-In Manager interface allows you to set the metaframe length.

Figure 1–27. Frame Generator



CRC-32 Generator

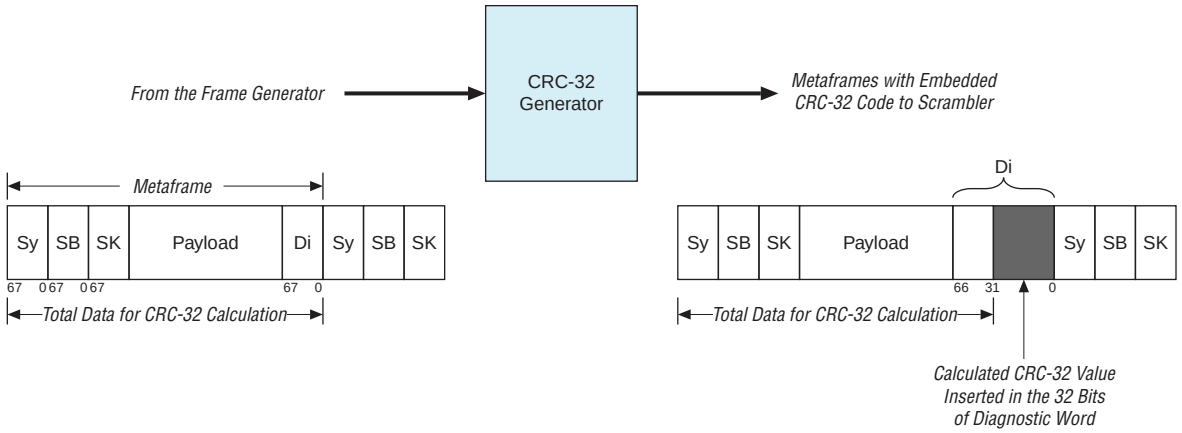
The CRC-32 generator is used only in Interlaken configurations.

The CRC-32 (Figure 1–28) generator block receives data from the frame generator and calculates the cyclic redundancy check (CRC) code for each block of data. This CRC code value is stored in the CRC32 field of the diagnostic word.

The CRC-32 calculation covers most of the metaframe, including the diagnostic word, except the following:

- bits [66:64] of each word
- 58-bit scrambler state within the scrambler state word
- 32-bit CRC-32 field within the diagnostic word

Figure 1–28. CRC-32 Generator



64B/66B Encoder

The 64B/66B encoder is used only in 10GBASE-R configurations.

The 64B/66B encoder conforms to the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49.

This block contains the 64B/66B encoder sub-block and the transmitter state machine sub-block. The 64B/66B encoder sub-block receives data from the transmitter FIFO and encodes the 64-bit data and 8-bit control characters to the 66-bit data block required by the 10GBASE-R configuration. The transmit state machine in the 64B/66B encoder sub-block checks the validity of the 64-bit data from the MAC layer and ensures proper block sequencing.

Scrambler

The scrambler operates in frame synchronous mode and self synchronous mode. Frame synchronous mode operates in Interlaken configurations. Self synchronous mode operates in 10GBASE-R configurations, as specified in IEEE 802.3-2008 clause-49.

Disparity Generator

The disparity generator is used only in Interlaken configurations.

The disparity generator block conforms to the Interlaken protocol specification and provides a DC-balanced data output. The disparity generator receives data from the scrambler and inverts the running disparity to stay within the ±96-bit boundary. To ensure this running disparity requirement, the disparity generator inverts bits [63:0] and sets bit [66] to indicate the inversion.

Table 1–12 interprets the MSB of the 67-bit payload.

Table 1–12. Interpretation of the MSB in the 67-Bit Payload for Stratix V Devices

MSB	Interpretation
0	Bits [63:0] are not inverted; the disparity generator processes the word without modification
1	Bits [63:0] are inverted; the disparity generator inverts the word before processing it

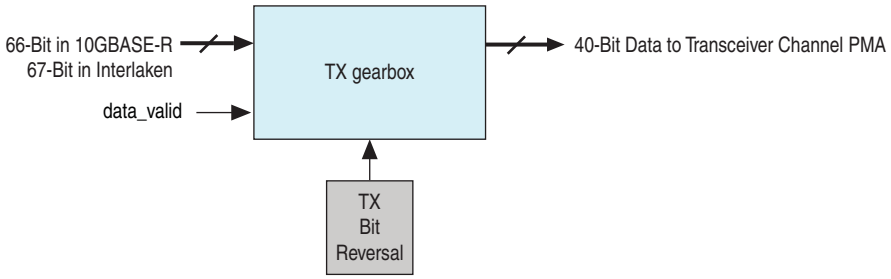
Transmitter Gearbox

The transmitter gearbox adapts the PCS data width to a smaller bus width for interfacing with the PMA. Because of the transmitter gearbox, the difference in the bus widths between the PCS and the PMA is transparent to the logic in the FPGA fabric.

For the supported PMA-PCS widths and for custom configurations, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

Figure 1–29 shows the transmitter gearbox with data widths for the Interlaken and 10GBASE-R configurations.

Figure 1–29. Transmitter Gearbox



In addition to providing bus width adaptation, the transmitter gearbox provides the transmitter polarity inversion, bit reversal and bit-slip features.

Transmitter Polarity Inversion

Transmitter polarity can be used to reverse the positive and negative differential buffer signals. This is useful if these signals are reversed on the board or backplane layout.

A high value on the `tx_invpolarity` register, which is accessed via the Avalon-MM PHY management interface, inverts the polarity of every bit of the input data word to the serializer in the transmitter datapath. Because inverting the polarity of each bit has the same effect as swapping the positive and negative signals of the differential link, correct data is sent to the receiver. Dynamically changing the `tx_invpolarity` register value might cause initial disparity errors at the receiver of an 8B/10B encoded link. The downstream system must be able to tolerate these disparity errors.

If polarity inversion is asserted midway through a serializer word, the word will be corrupted.

Transmitter Bit Reversal

The transmitter gearbox can reverse the order of transmitted bits. By default, the transmitter first sends out the LSB of a word. Some protocols, such as Interlaken, require that the MSB of a word (bit 66 in a word `[66:0]`) is transmitted first. When you enable the transmitter bit reversal, the parallel input to the gearbox is swapped and the MSB is sent out first. The Quartus II software automatically sets the bit reversal for Interlaken configurations.

Transmitter Bit-Slip

The transmitter bit-slip allows you to compensate for the channel-to-channel skew between multiple transmitter channels by slipping the data sent to the PMA. The maximum number of bits slipped is controlled from the FPGA fabric and is equal to the width of the PMA-PCS interface, minus one.

PCIe Gen 3 PCS Architecture

Stratix V supports the PCIe Gen 3 Base specification. The PCIe Gen 3 uses a 128/130 bit block encoding/decoding scheme which is different from the 8B/10B scheme used in Gen 1 and Gen 2. The 130-bit block contains a 2-bit sync header and 128-bit data payload. For this reason, the PCIe Gen 3 PCS has a separate data path as compared to the PCIe Gen 1 or Gen 2 PCS. The PCIe Gen 3 PCS supports the PHY Interface for the PCI Express (PIPE) interface with the hard IP (HIP) enabled and with the HIP bypassed.

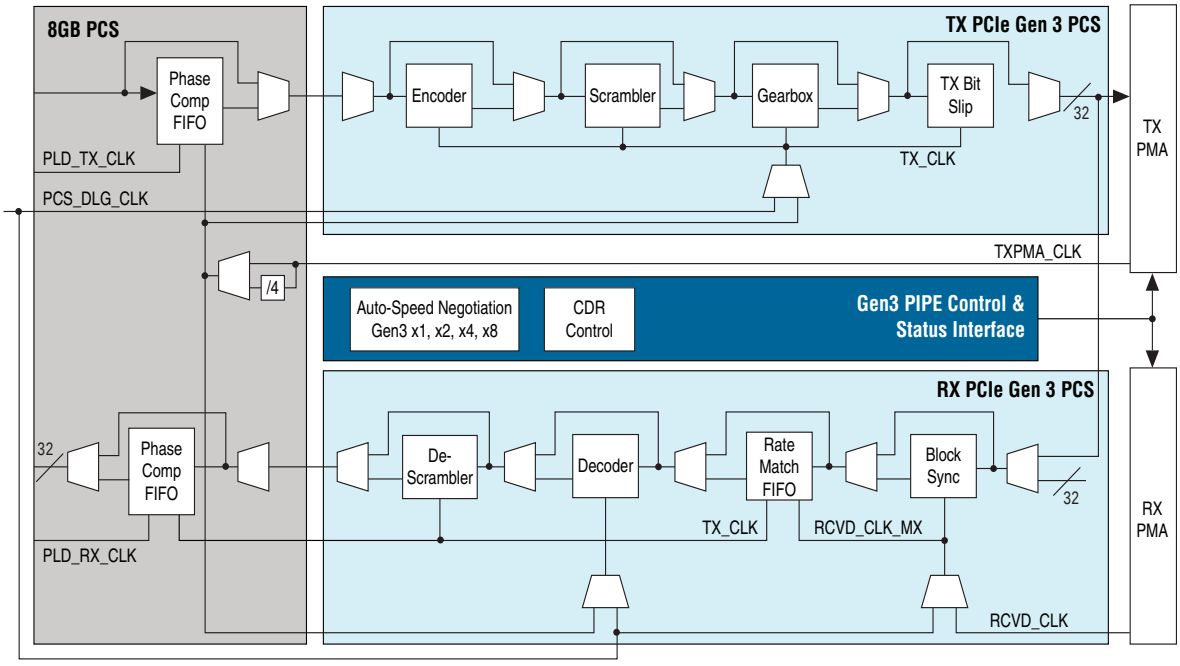
This section contains the following topics:

- "Receiver PCIe Gen 3 PCS Datapath" on page 1–43
- "Transmitter PCIe Gen 3 PCS Datapath" on page 1–44
- "PIPE Interface" on page 1–44

This PIPE interface supports the seamless switching of Data and Clock between the Gen 1, Gen 2, and Gen 3 PCS, and provides support for PIPE 3.0 features.

The overall simplified PCIe Gen 3 PCS architecture diagram is in Figure 1–30. Note that the RX/TX Phase Comp FIFOs are physically placed in, and shared with the standard 8GB PCS.

Figure 1–30. PCIe Gen 3 PCS Top Level Block Diagram



Receiver PCIe Gen 3 PCS Datapath

This section describes the receiver channel PCIe Gen 3 PCS datapath architecture. Figure 1–30 shows the receiver GX channel datapath for the PCIe Gen 3 PCS.

Block Sync

PMA parallelization occurs at arbitrary word boundaries. Consequently, the parallel data from the RX PMA CDR needs to be realigned to meaningful character boundaries. The Block Sync module searches for the Electrical Idle Exit Sequence Ordered Set (or the last number of fast training sequences (NFTS) Ordered Set) and skip (SKP) Ordered Set to identify the correct boundary for the incoming stream and achieve the block alignment. The block is realigned to the new block boundary following the receipt of a SKP Ordered Set, as it can be of variable length.

Rate Match FIFO

The Rate Match FIFO (or clock compensation FIFO) compensates for minute frequency differences between the local clock (sometimes referred to as PLD soft IP clock, or PLD system clock) and the recovered clock. This is achieved by inserting and deleting SKP characters in the data stream to keep the FIFO from going empty or full respectively.

Decoder

The Decoder checks for decode errors in the data stream. It also enables or disables the Descrambler based on the Data and Ordered Set received.

Descrambler

The Descrambler descrambles data per the PCIe Gen 3 specification.

Phase Compensation FIFO (Shared with Standard 8GB PCS)

Refer to the "Receiver Phase Compensation FIFO" on page 1–30

Transmitter PCIe Gen 3 PCS Datapath

This section describes the transmitter channel PCIe Gen 3 PCS datapath architecture. Figure 1–30 shows the transmitter GX channel datapath for the PCIe Gen 3 PCS.

Phase Compensation FIFO (Shared with Standard PCS)

Refer to the "Receiver Phase Compensation FIFO" on page 1–30.

Scrambler

The Scrambler is an additive scrambler on a per-lane basis with degree 23 polynomial linear feedback shift register (LFSR), with different taps for the 8 adjacent lanes. The scrambler is used to provide enough edge density, since there is no 8B/10B encoding in PCIe Gen 3, so that the RX PMA CDR can lock to the incoming data stream and generate the recovered clock.

Encoder

The PCIe Gen 3 base specification defines that the data packets have to be scrambled and descrambled, whereas the Ordered Set packets (except the first symbol of TS1 and TS2 Ordered Set) do not have to be scrambled or descrambled. The Encoder/Decoder continuously checks the header and payload of the packet and generates a signal to enable the scrambler/descrambler based upon whether the payload is an ordered set, or data packet. It also generates a signal to reset the scrambler/descrambler to the initial seed value if an Electrical Idle Exit Ordered Set or a Fast Training Sequence Ordered Set is received or transmitted. In addition, the encoder/decoder logic monitors the Ordered Set and the header for invalid values, and generates an error flag if they do.

Gearbox

The PCIe 3.0 base specification specifies a block size of 130 bits, with the exception of SKP Ordered Sets which can be variable length. An implementation of a 130-bit data path would take up significant resources, so the PCIe Gen 3 PCS data path is implemented as 32 bits wide. As the TX PMA data width is fixed to 32 bits, and the block size is 130 bits with variations, a gearbox is needed to convert the 130 bits to 32 bits. This gearbox has a transmitter bit-slip feature.

PIPE Interface

The PIPE Interface block provides PIPE Gen 3 compliant control and status interfaces between the high-speed serial interface (HSSI) and upper layer logic. It is PIPE 3.0 compliant.

Auto Speed Negotiation Block

The Auto Speed Negotiation block controls the operating speed of the HSSI when operating under PIPE 3.0 modes. By monitoring the rate control signal from the PhyMac, this block will change the HSSI from PCIe Gen 1 operation mode, to Gen 2 operation mode, or from PCIe Gen 1 operation mode to Gen 2 operation mode to Gen 3 operation mode, or vice versa, with all the appropriate settings.

Electrical Idle Inference Block

In conjunction with side band signals from the PLD side, the Electrical Idle Inference Block infers Electrical Idle assuming that the signal detect is not reliable. This is based on the PCIe Base Specification Revision 2.0/3.0.

Clock Data Recovery (CDR) Control Block

The CDR control block is used for Rx.L0s fast exit when operating in PIPE/PCIe Gen 3 mode. Upon detecting an Electrical Idle Ordered Set (EIOS), it takes manual control of the CDR by forcing it into a lock to reference mode. When an exit from electrical idle is detected, this block moves the CDR into lock to data mode to achieve fast data lock.

Document Revision History

Table 1–13 lists the revision history for this chapter.

Table 1–13. Document Revision History (Part 1 of 2)

Date	Version	Changes
June 2012	2.3	■ Updated "Stratix V Device Layout", "PMA Architecture", "Standard PCS Architecture" and "10G PCS Architecture" sections. ■ Updated Table 1–2, Table 1–4, Table 1–1, and Table 1–5. ■ Updated Figure 1–1, Figure 1–3, Figure 1–4, Figure 1–8, and Figure 1–21. ■ Updated "Transmitter Polarity Inversion" section. ■ Added "PCIe Gen 3 PCS Architecture" section.
February 2012	2.2	■ Updated Figure 1–1. ■ Updated "Transmitter Polarity Inversion" section.
December 2011	2.1	■ Updated Table 1–5, Table 1–6, Table 1–7, Table 1–13, Table 1–17. ■ Updated Figure 1–1, Figure 1–2, Figure 1–3, Figure 1–6, Figure 1–12, Figure 1–18, Figure 1–21, Figure 1–22. ■ Updated "Stratix V Device Layout", "PMA Architecture", "Standard PCS Architecture", "10G PCS Architecture", "Bonded Configurations" sections. ■ Added the "Deskew FIFO" section.

Table 1–13. Document Revision History (Part 2 of 2)

Date	Version	Changes
November 2011	2.0	■ Updated Table 1–1, Table 1–3, and Table 1–5. ■ Updated Figure 1–4, Figure 1–5, Figure 1–6, Figure 1–7, Figure 1–8, Figure 1–9, Figure 1–10, Figure 1–13, Figure 1–14, and Figure 1–15. ■ Updated the "Receiver Clock Data Recovery Unit", "Receiver Buffer", "Receiver Analog Settings", "Link Coupling", "Byte Serializer", "8B/10B Encoder", "Receiver FIFO", "PCIe Receiver Detect", "Programmable Pre-Emphasis". "Signal Threshold Detection Circuitry", "Offset Cancellation in the Receiver Buffer and Receiver CDR", "Channel PLL Used as a CMU PLL (Transmitter PLL)", "Auxiliary Transmit (ATX) PLL Architecture", "Serializer", "Deserializer", "Transmitter Buffer", "Transceiver Calibration Blocks", and "10G PCS Architecture" sections. ■ Added the "Continuous Time Linear Equalization". "EyeQ", "XAUI Mode", "PCIe Mode", and "PMA Direct Mode" sections. ■ Added Stratix V GT channel information. ■ Added Table 1–2. ■ Added Figure 1–2 and Figure 1–19. ■ Removed Table 1-13. ■ Removed Figure 1-13. ■ Minor text edits.
May 2011	1.4	■ Chapter moved to Volume 3. ■ Minor text edits.
December 2010	1.3	■ Updated Figure 1–16. ■ Minor graphic edits
December 2010	1.2	■ Updated to include Quartus II version 10.1 information. ■ Reorganized chapter information. ■ Added the "Channel PLL Used as a CMU PLL (Transmitter PLL)" and "Auxiliary Transmit (ATX) PLL Architecture" sections. ■ Added Figure 1–11. ■ Updated the "Bonded Configuration" section. ■ Updated Figure 1–2, Figure 1–3, Figure 1–4, Figure 1–5, Figure 1–7, Figure 1–8, Figure 1–14, Figure 1–16, and Figure 1–29. ■ Reversed Rx directional flow in Figure 1–8, Figure 1–18, Figure 1–19, Figure 1–20, Figure 1–21, and Figure 1–22. ■ Minor text edits.
July 2010	1.1	Initial release in the *Stratix V Device Handbook*.
April 2010	1.0	Initial release for EAP.

2. Transceiver Clocking in Stratix V Devices

SV52003-1.6

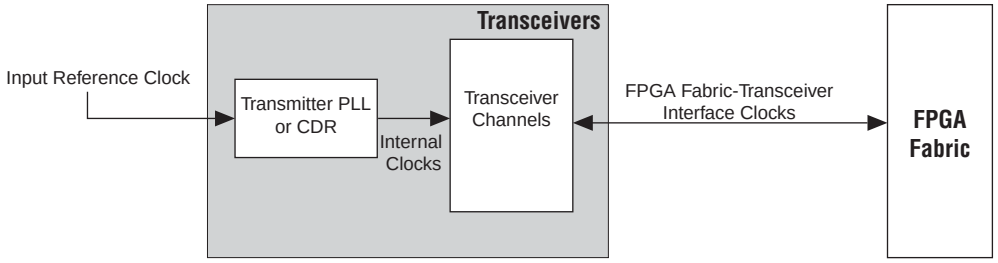
This chapter provides detailed information about the Stratix® V transceiver clocking architecture.

The clocking architecture chapter is divided into three sections:

- "Input Reference Clocking"—Describes how the reference clock to the transmitter phase-locked loop (PLL) and clock data recovery (CDR) generates the clocks required for transceiver operation.
- "Internal Clocking" on page 2–5—Describes the clocking architecture internal to the transceiver.
- "FPGA Fabric-Transceiver Interface Clocking" on page 2–29—Describes the clocking options available when the transceiver interfaces with the FPGA fabric.

Figure 2–1 shows an overview of the transceiver clocking architecture.

Figure 2–1. Transceiver Clocking Architecture Overview



For upcoming clocking related features, refer to the *Upcoming Stratix V Device Features* document.

Input Reference Clocking

Each transceiver channel has a channel PLL that can be configured as a transmitter clock multiplier unit (CMU) PLL or a receiver CDR PLL. In the CMU PLL configuration, the channel PLL uses the input reference clock to generate a serial clock. In the receiver CDR PLL configuration, the channel PLL locks to the input reference clock in lock-to-reference (LTR) mode. The auxiliary transmit (ATX) PLL uses the input reference clock to synthesize a serial clock.

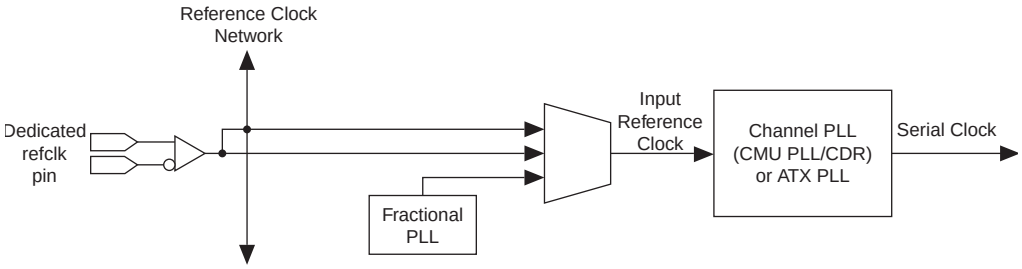
© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Input Reference Clock Sources

The channel PLL and ATX PLL can derive the input clock from a dedicated `refclk` pin, a fractional PLL, or through the reference clock network. Figure 2–2 shows an overview of the input reference clock sources to the channel PLL or ATX PLL.

Figure 2–2. Input Reference Clock Sources to Channel PLL or ATX PLL



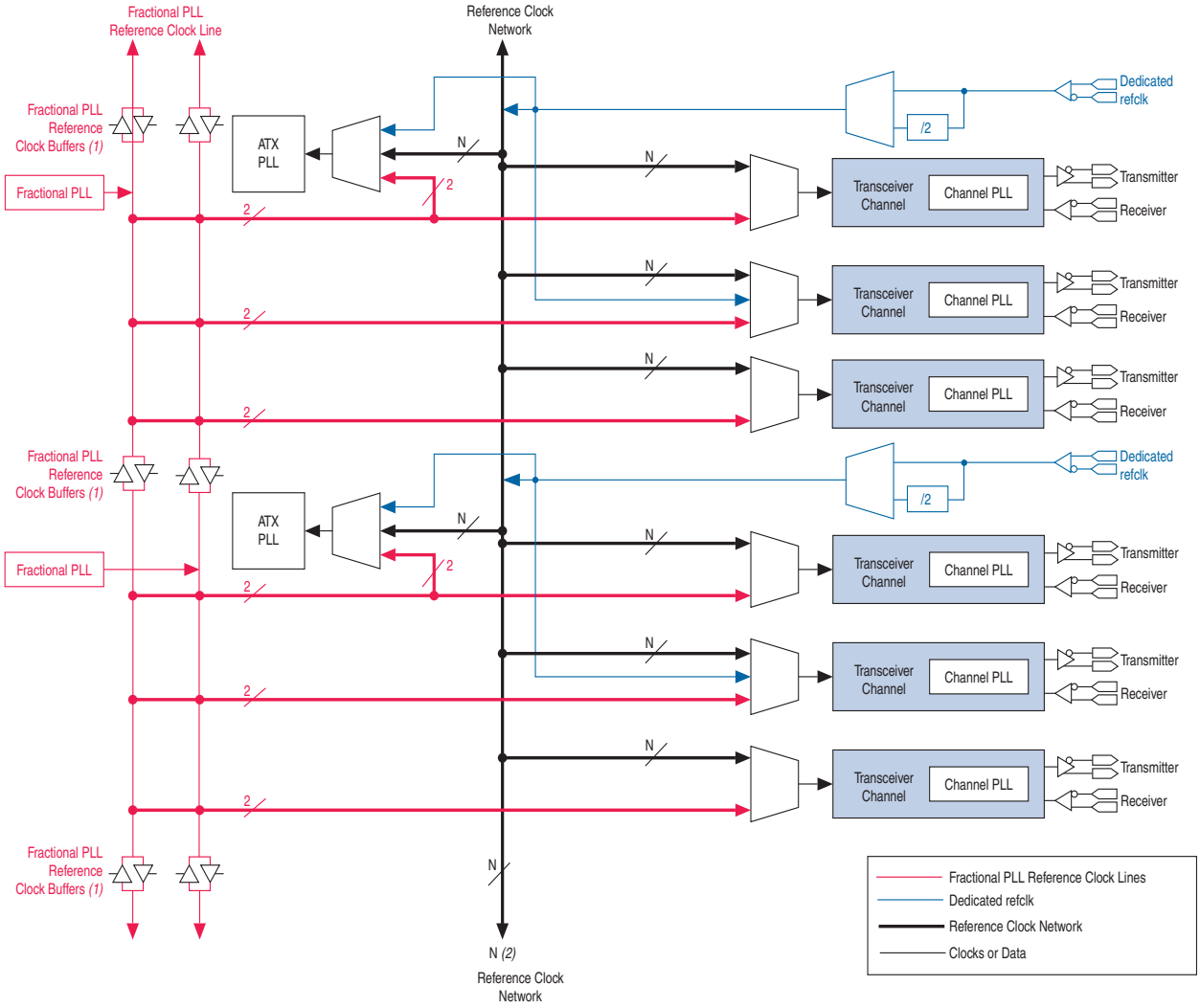
The following input reference clock resources for transmitter PLLs are listed in the order of jitter performance, beginning with the best input reference clock source:

- "Dedicated refclk Pins" on page 2–4
- "Dedicated refclk Pins Using the Reference Clock Network" on page 2–5
- "Fractional PLLs" on page 2–5 (in integer mode and fractional mode)

For optimal performance, use the `refclk` source that is closest to the transmit PLL in the same transceiver bank.

Figure 2–3 shows the input reference clock sources for GX transceiver channels in a GX transceiver bank.

Figure 2–3. Input Reference Clock Sources for GX Transceiver Channels



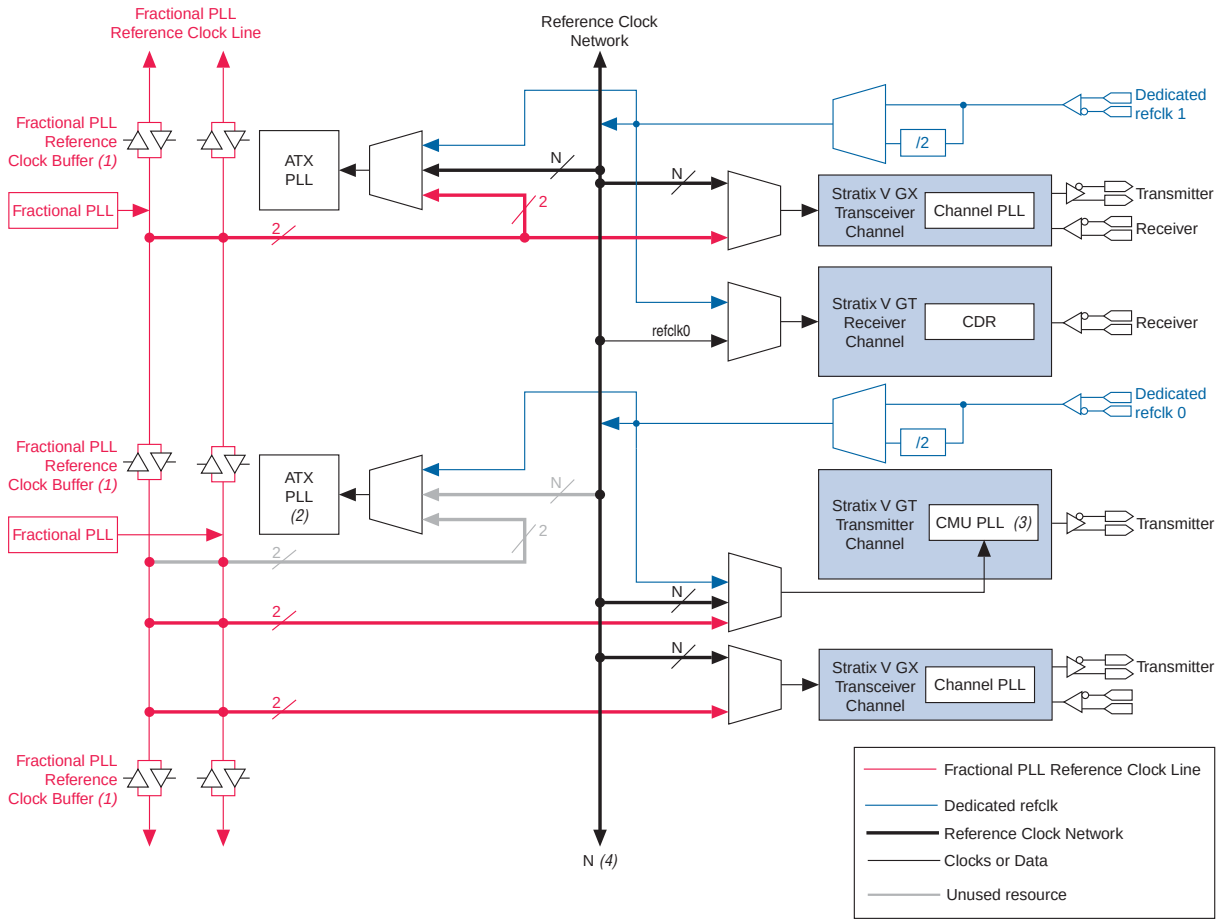
Notes to Figure 2–3:

(1) The fractional PLL refclk buffers allow you to segment the reference clock line into multiple segments, such that fractional PLLs in different transceiver banks can drive the same fractional PLL reference clock line.

(2) `N` equals the number of transceiver channels on a side divided by 3, which is equal to the number of dedicated `refclk` pins.

Figure 2–4 shows the input reference clock sources for a GT transceiver channel and two GX transceiver channels in a GT transceiver bank.

Figure 2–4. Input Reference Clock Sources for GT and GX Transceiver Channels



Notes to Figure 2–4:

(1) The fractional PLL refclk buffers allow you to segment the reference clock line into multiple segments, such that fractional PLLs in different transceiver banks can drive the same fractional PLL reference clock line.

(2) The bottom ATX PLL of a GT transceiver bank provides the serial clock to the GT transmitter channel.

(3) The CMU PLL of the GT transmitter channel drives an x1 clock line that can be used by the top and bottom GX transceiver channels in the GT transceiver bank.

(4) N equals twice the number of GT channels.

Altera recommends using a dedicated clock `refclk 0` for the bottom ATX PLL that provides the serial clock to the GT transmitter channel.

Dedicated refclk Pins

GX transceiver banks have one dedicated `refclk` pin for each group of three transceiver channels. The dedicated `refclk0/refclk1` pins can drive reference clock network or ch1/ch4 channel plls respectively in a transceiver bank.

There are two dedicated `refclk` pins available in each GT transceiver bank. The two `refclk` pins can also provide the reference clocks to the GX channels in a GT transceiver bank through the reference clock network.

Dedicated refclk Pins Using the Reference Clock Network

Each dedicated `refclk` pin can drive any transmitter PLL on the same side of the device through the reference clock network. Designs using multiple transmitter PLLs that require the same reference clock frequency and are located along the same side of the device can share the same dedicated `refclk` pin.

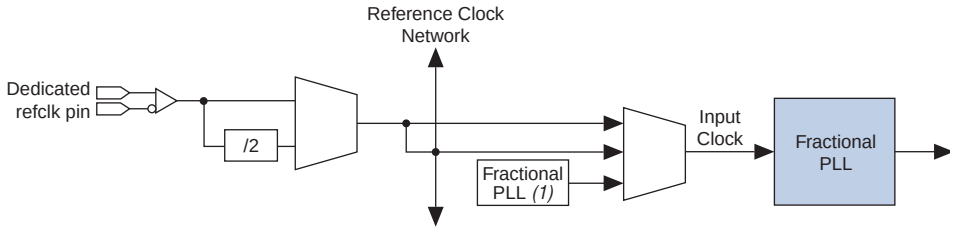
Fractional PLLs

Stratix V devices provide a fractional PLL for each group of three transceiver channels. Each fractional PLL drives one of two clock lines spanning the side of the device that can provide an input reference clock to any transmitter PLL or CDR on the same side of the device. A fractional PLL enables you to use an input reference clock in your system that is not supported by the transmitter PLL or CDR to synthesize a supported input reference clock. Fractional PLLs support fractional and integer modes. Fractional mode allows you to synthesize a clock of any supported frequency while integer mode allows you to synthesize an output clock that is an integer multiple or factor of the input clock. For example, fractional PLLs allow you to take a 100 MHz clock and synthesize a clock of 50 or 200 MHz in integer mode or 614.4 MHz in fractional mode.

The jitter performance is best with ATX PLL s, then CMU PLLs, and then fractional PLLs.

Figure 2–5 shows the input clock sources for the fractional PLL.

Figure 2–5. Fractional PLL Input Clock Sources



Note to Figure 2–5:

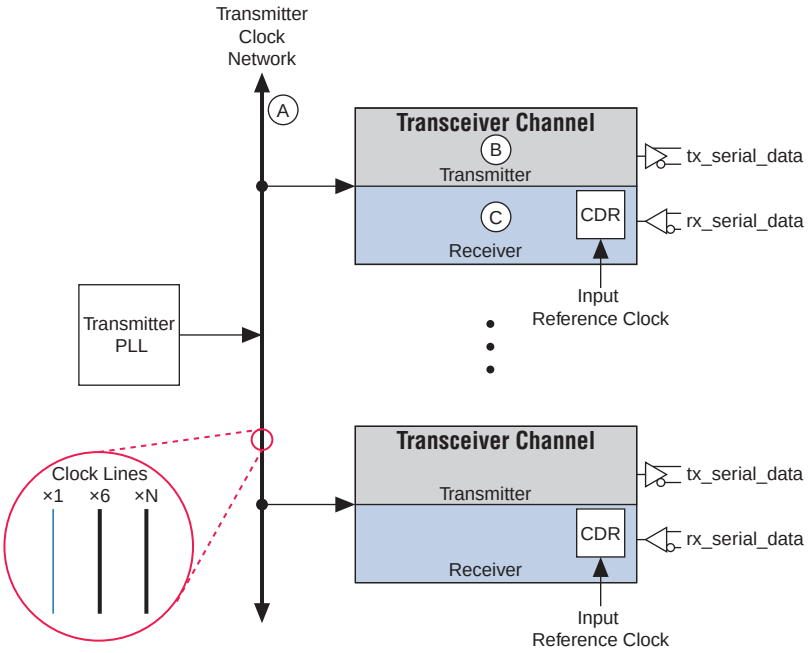
(1) A fractional PLL can provide a clock source to another fractional PLL through the reference clock network driven by the fractional PLLs. However, this PLL cascading can result in higher jitter.

Internal Clocking

This section describes the clocking architecture internal to Stratix V transceivers. Different physical coding sublayer (PCS) configurations and channel bonding options result in different transceiver clock paths. Figure 2–6 shows the following sections of the transceiver internal clocking:

- "Transmitter Clock Network" (A in Figure 2–6)
- "Transmitter Clocking" (B in Figure 2–6)
- "Receiver Clocking" (C in Figure 2–6)

Figure 2–6. Internal Clocking



The reference clock from one of the sources shown in Figure 2–2 is fed to a transmitter PLL. The transmitter PLL could be either a channel PLL configured as a CMU PLL or an ATX PLL. The transmitter PLL generates a serial clock that is distributed using a transmitter clock network to the transceiver channels.

The clocking described in this section is internal to the transceiver, and the clock routing is primarily performed by the Quartus II software, based on the transceiver configuration selected.

Transmitter Clock Network

The transmitter clock network routes the clock from the transmitter PLL to the transmitter channel (as shown in Figure 2–6) and provides two clocks to the transmitter channel:

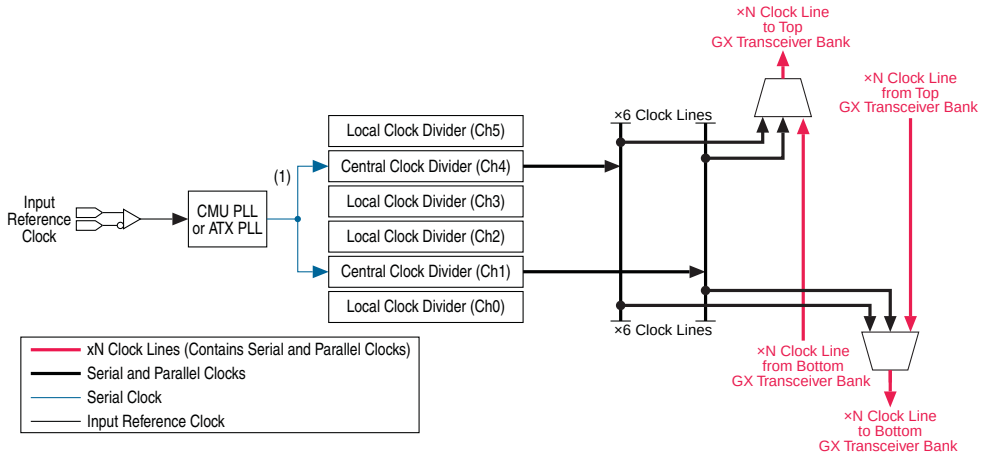
- Serial clock—high-speed clock for the serializer
- Parallel clock—low-speed clock for the serializer and the PCS

Stratix V transceivers support various non-bonded and bonded transceiver clocking configurations. If you use a bonded configuration, both the serial clock and parallel clock are routed from the transmitter PLL to the transmitter channel. If you use a non-bonded configuration, then only the serial clock is routed from the transmitter PLL to the transmitter channel and the parallel clock is generated by the clock divider of each channel.

For more information about bonding, refer to the "Bonded Configurations" section of the *Transceiver Architecture in Stratix V Devices* chapter.

Figure 2–7 shows the transceiver clock network, beginning with the input reference clock, followed by the transmitter PLL, clock dividers, and ending with the x6 and xN clock lines.

Figure 2–7. Transmitter Clock Network


×N Clock Line to Top GX Transceiver Bank
×N Clock Line from Top GX Transceiver Bank
×6 Clock Lines
Local Clock Divider (Ch5)
Central Clock Divider (Ch4)
Local Clock Divider (Ch3)
Local Clock Divider (Ch2)
Central Clock Divider (Ch1)
Local Clock Divider (Ch0)
(1)
Input Reference Clock
CMU PLL or ATX PLL
×6 Clock Lines
×N Clock Line from Bottom GX Transceiver Bank
×N Clock Line to Bottom GX Transceiver Bank
xN Clock Lines (Contains Serial and Parallel Clocks)
Serial and Parallel Clocks
Serial Clock
Input Reference Clock

Note to Figure 2–7:

(1) This can be either an x1 clk line or a direct path from the TX PLL to the clock divider.

Transmitter Clock Lines

The transmitter clock network consists of two types of dedicated clocking resources:

- Non-bonded configurations
  - x1 clock lines
- Bonded configurations (not available for GT transceiver channels)
  - x6 clock lines
  - xN clock lines (available only for PCIe)

The Quartus II software performs the clock routing related to the transmitter clock network based on the transceiver configuration selected.

Table 2–1 lists the span and data rates supported by clock sources and networks in Stratix V devices.

Table 2–1. Data Rates and Spans Supported Using Stratix V Clock Sources and Clock Networks

Clock Network	Transceiver Channel	Clock Source	Max Data Rate	Bonding	Span
x1	GX	■ Ch1 or Ch4 CMU PLL in a transceiver bank ■ ATX PLLs in a transceiver bank	14.1 Gbps (1)	No	Transceiver bank
	GT	Bottom ATX PLL in a GT transceiver bank	28 Gbps (1)	No	Transceiver bank
x6	GX	Central clock dividers in a transceiver bank (in Ch1 or Ch4 only)	14.1 Gbps (1)	Yes	Transceiver bank
xN	PCIe only (2)	The channel PLL provides a serial clock to the central clock dividers of Ch1 and Ch4. The central clock dividers in the transceiver bank drive the x6 clock lines. The xN clock lines receive the serial and parallel clocks from the x6 clock lines. Refer to Figure 2–7.	8 Gbps	Yes	xN lines span a side of the device, but can bond only up to eight contiguous data channels

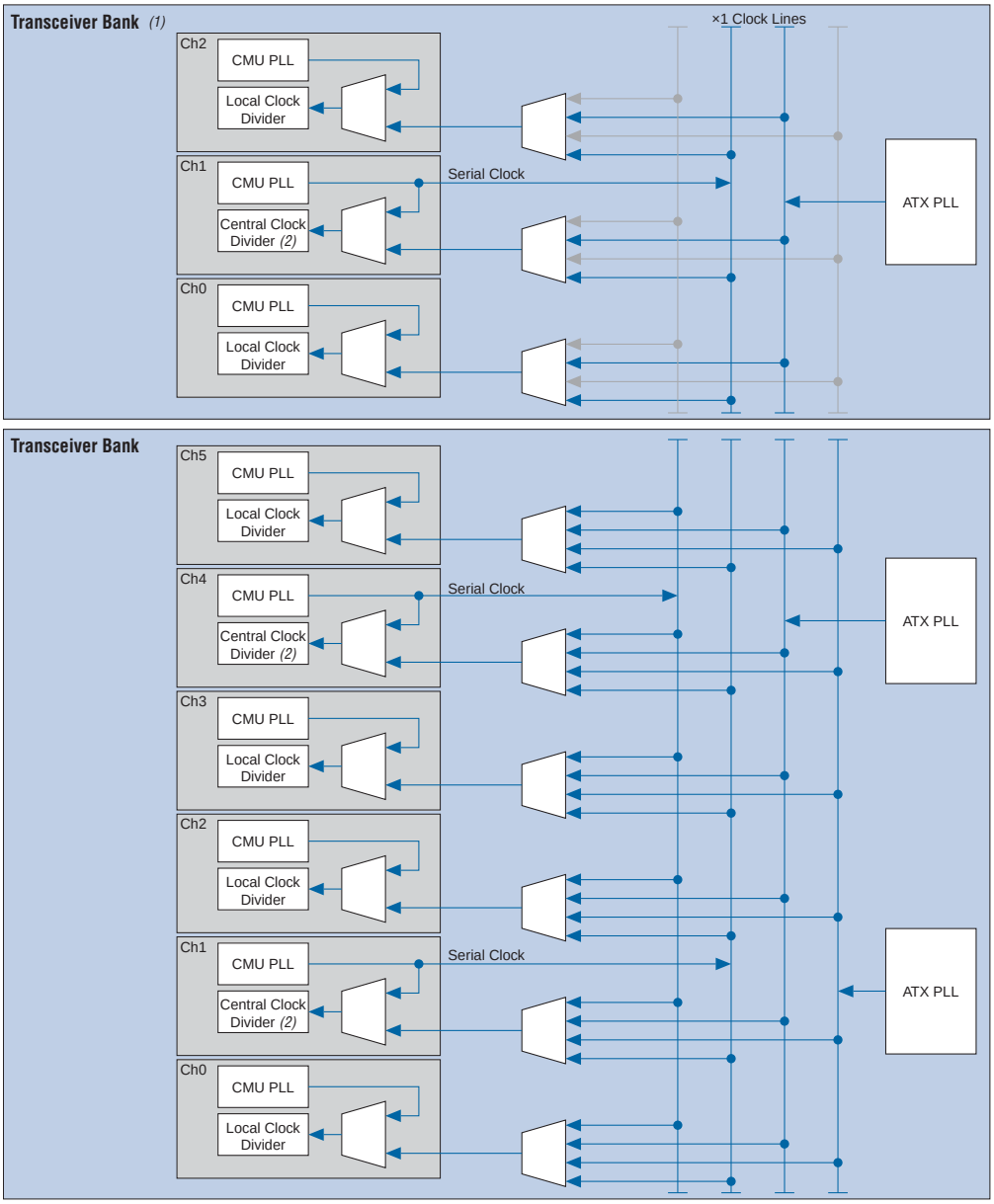
Notes to Table 2–1:

(1) For the fastest speed grade only. For the remaining speed grades, refer to the *Stratix V Device Datasheet*.

(2) For more information about PCIe x8 configurations, refer to the section titled "Hard IP x8 Configuration" in the *Transceiver Configurations in Stratix V Devices* chapter.

Figure 2–8 shows the x1 clock lines used by GX transmitter channels.

Figure 2–8. x1 Clock Lines Used by GX Transmitter Channels for Non-Bonded Configuration



Notes to Figure 2–8:

(1) Stratix V devices 5SGXB5, 5SGXB6, 5SGSB, and 5SGSB have one transceiver bank on each side with only three transceiver channels. For more information, refer to the *Transceiver Architecture in Stratix V Devices* chapter.

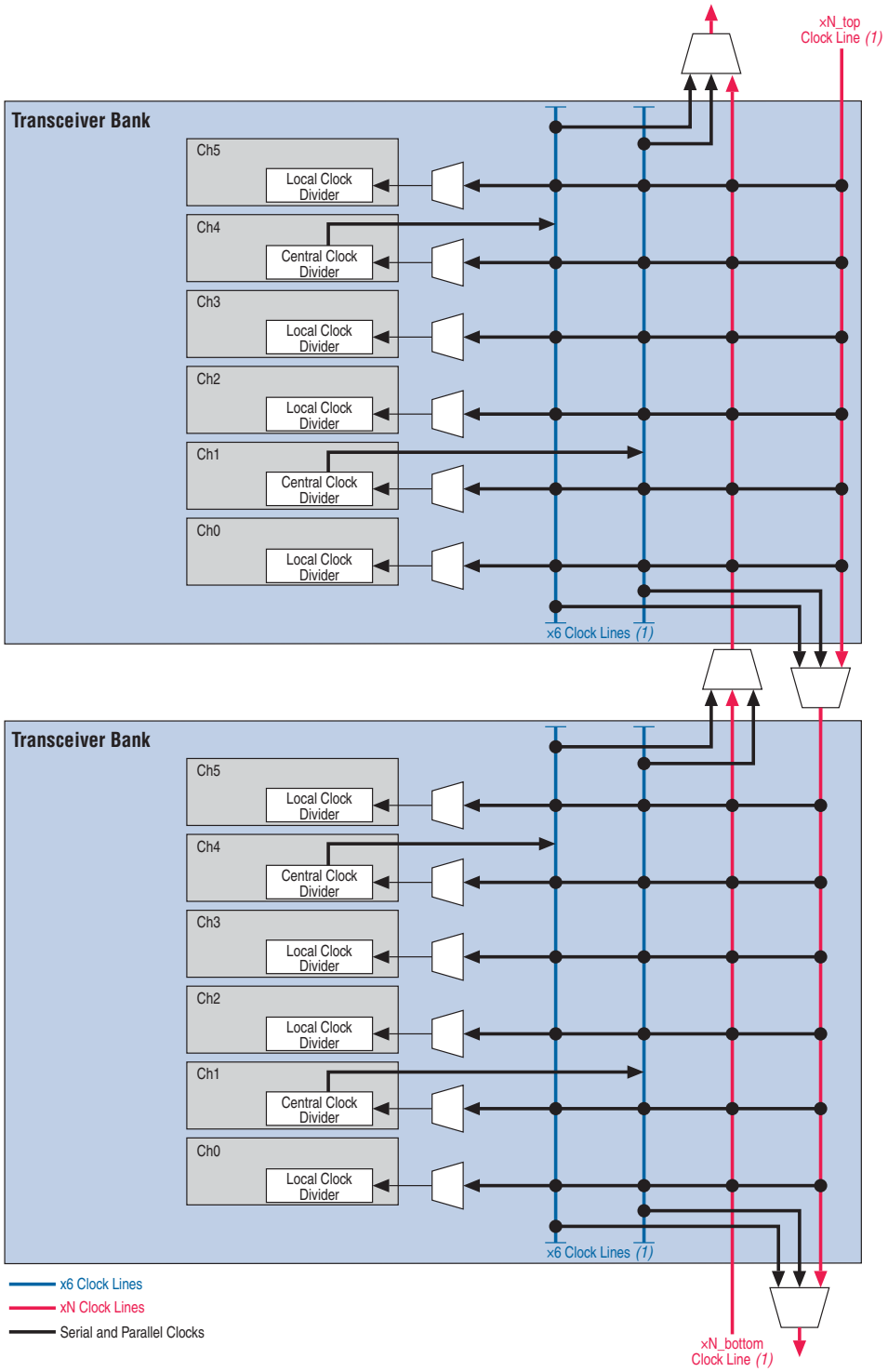
(2) You can use the central clock divider as a local clock divider.

The x1 clock lines route the serial clock to the clock dividers of any channel within a transceiver bank. Refer to Table 2–1 on page 2–8 for details. The channel PLL, if configured as a CMU PLL, can drive the clock divider of its own channel, but you will not be able to use the channel PLL as a CDR. Without a CDR, you can use the channel only as a transmitter channel.

The x6 and xN clock lines are used for bonded configurations and route both the serial clock and parallel clock from the central clock dividers to the transceiver channels.

Figure 2–9 shows x6 and xN clock lines. The central clock dividers of channel 1 and channel 4 in a transceiver bank drive the x6 clock lines. The x6 clock lines then drive the xN clock lines. The xN clock lines, used only for PCIe x8 configurations, span the entire side of the device and can provide the serial and parallel clock to any eight contiguous channels within or outside a transceiver bank.

Figure 2–9. x6 and xN Clock Lines Used for Bonded Configurations



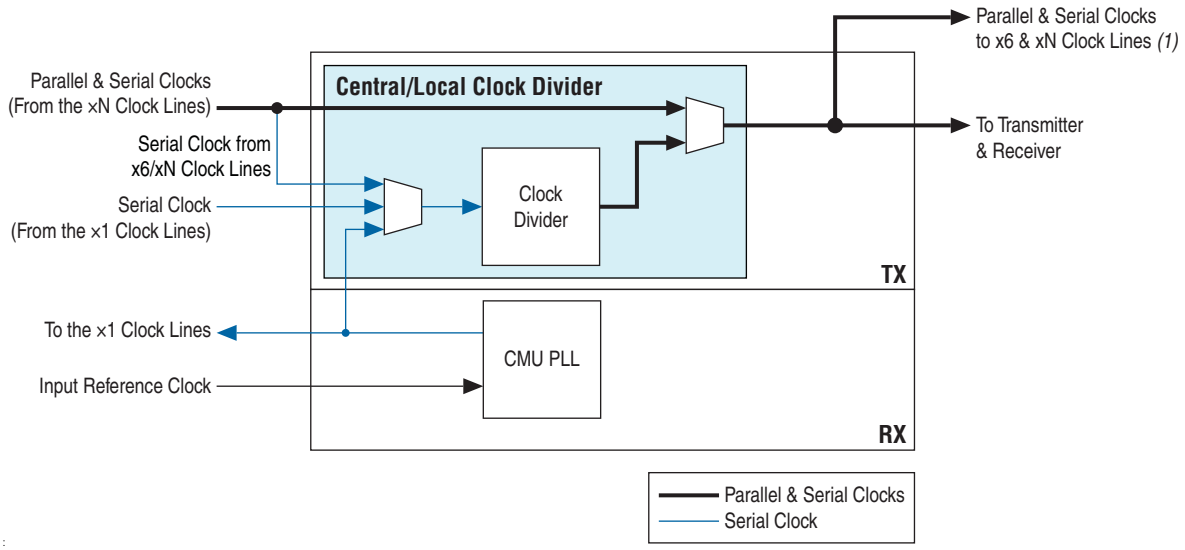
Note to Figure 2–9:

(1) The clock lines carry both serial and parallel clocks.

Clock Dividers

Each transmitter channel has a local clock divider. Some of the clock dividers that drive the x6 and xN clock lines are called central clock dividers. Figure 2–10 shows these two types of clock dividers. Central clock dividers are located in channels 1 and 4 of the GX transceiver bank. The clock dividers generate the parallel and serial clock sources for the transmitter and optionally for the receiver PCS. The central clock dividers can feed the clock lines used to bond channels.

Figure 2–10. Clock Dividers


Parallel & Serial Clocks to x6 & xN Clock Lines (1)
Parallel & Serial Clocks (From the xN Clock Lines)
Central/Local Clock Divider
Serial Clock from x6/xN Clock Lines
Serial Clock (From the x1 Clock Lines)
Clock Divider
To Transmitter & Receiver
TX
To the x1 Clock Lines
CMU PLL
Input Reference Clock
RX
Parallel & Serial Clocks
Serial Clock

Note to Figure 2–10:

(1) This is available only for the central clock divider in channels 1 and 4.

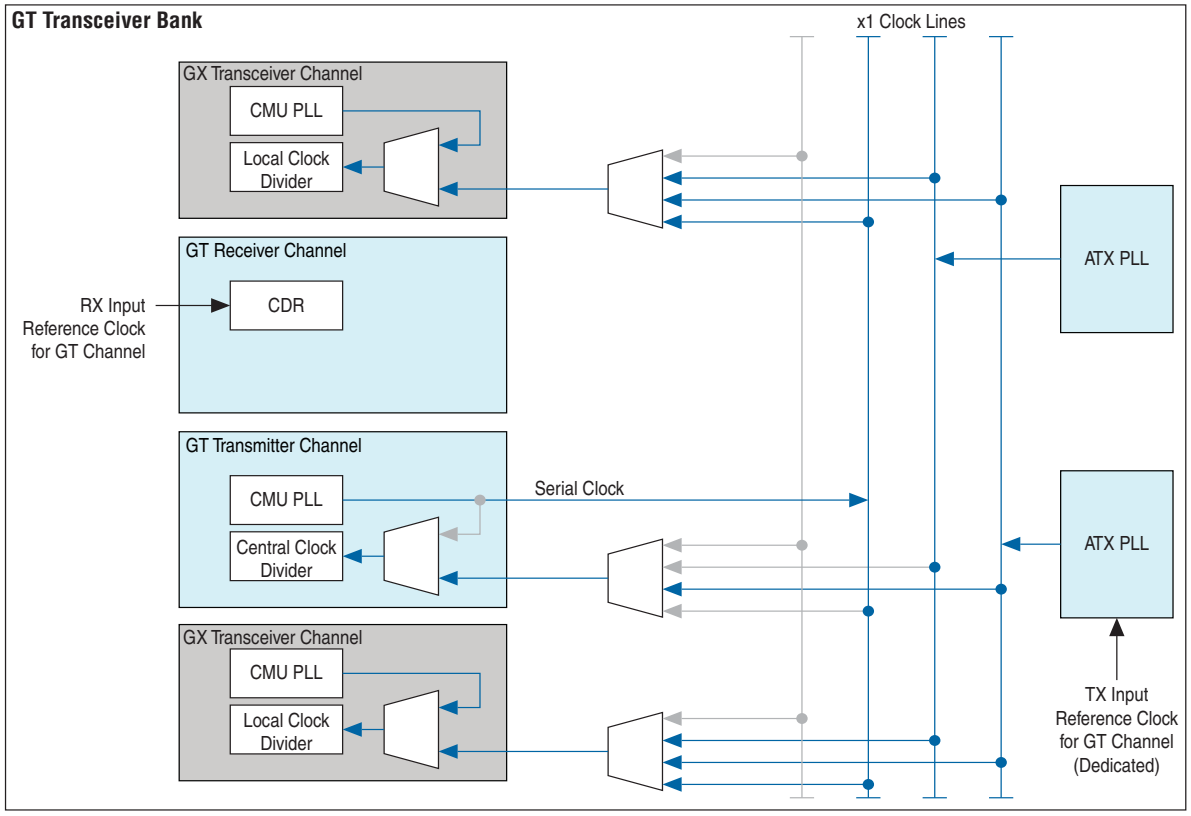
Transmitter Clock Network in Stratix V GT Transceiver Channels

The x1 clock lines in GT transceiver banks route the serial clock from the ATX PLL to the central clock divider of the GT transmitter channel. The x1 clock lines can also route the serial clock from the GT transmitter channel CMU PLL and the ATX PLLs to the local clock dividers of the GX transceiver channels.

Figure 2–11 shows the x1 clock lines used by a GT transceiver channel and two GX transceiver channels in a GT transceiver bank. There is one GT transmitter channel per GT transceiver bank. The transmitter channel must receive the clock from the bottom ATX PLL of a GT transceiver bank.

The channel PLL in the GT receiver channel is always used as a CDR.

Figure 2–11. x1 Clock Lines Used by Stratix V GT Transmitter Channels



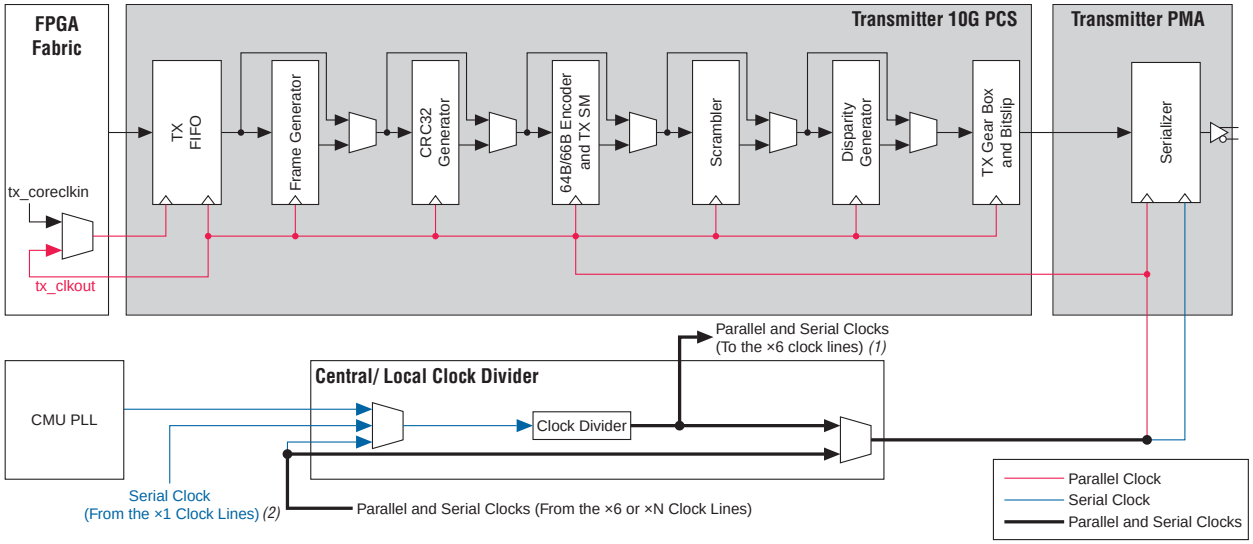
Transmitter Clocking

Transmitter clocking refers to the clocking architecture internal to the transmitter channel of a transceiver. This section discusses clocking for GX and GT transmitter channels.

Transmitter 10G PCS Clocking

Figure 2–12 shows the clocking for the transmitter 10G PCS and transmitter physical medium attachment (PMA).

Figure 2–12. Transmitter 10G PCS Clocking



Notes to Figure 2–12:

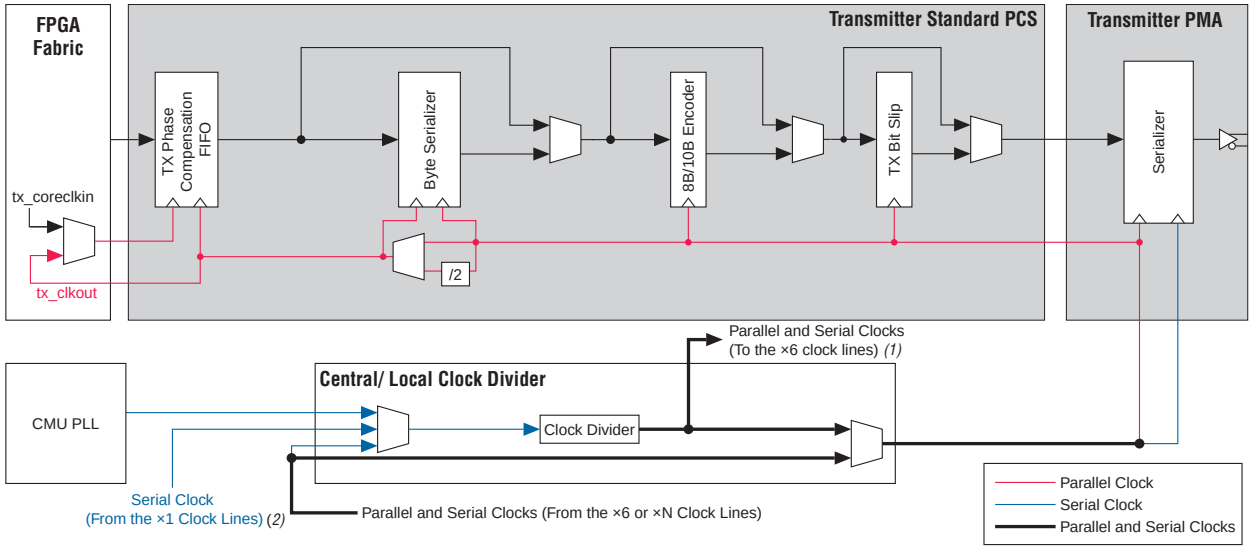
(1) Available only in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

(2) x1 clock lines can be driven by either a CMU PLL or ATX PLL.

Transmitter Standard PCS Clocking

Figure 2–13 shows the clocking for the transmitter standard PCS and transmitter PMA.

Figure 2–13. Transmitter Standard PCS Clocking



Notes to Figure 2–13:

(1) Available only in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

(2) x1 clock lines can be driven either by a CMU PLL or ATX PLL.

As shown in Figure 2–12 and Figure 2–13, the clock divider block provides the serial clock to the serializer of the transmitter PMA and the parallel clock to the transmitter PCS.

In the 10G PCS channel, the parallel clock is used by all the blocks up to the read side of the transmitter (TX) FIFO.

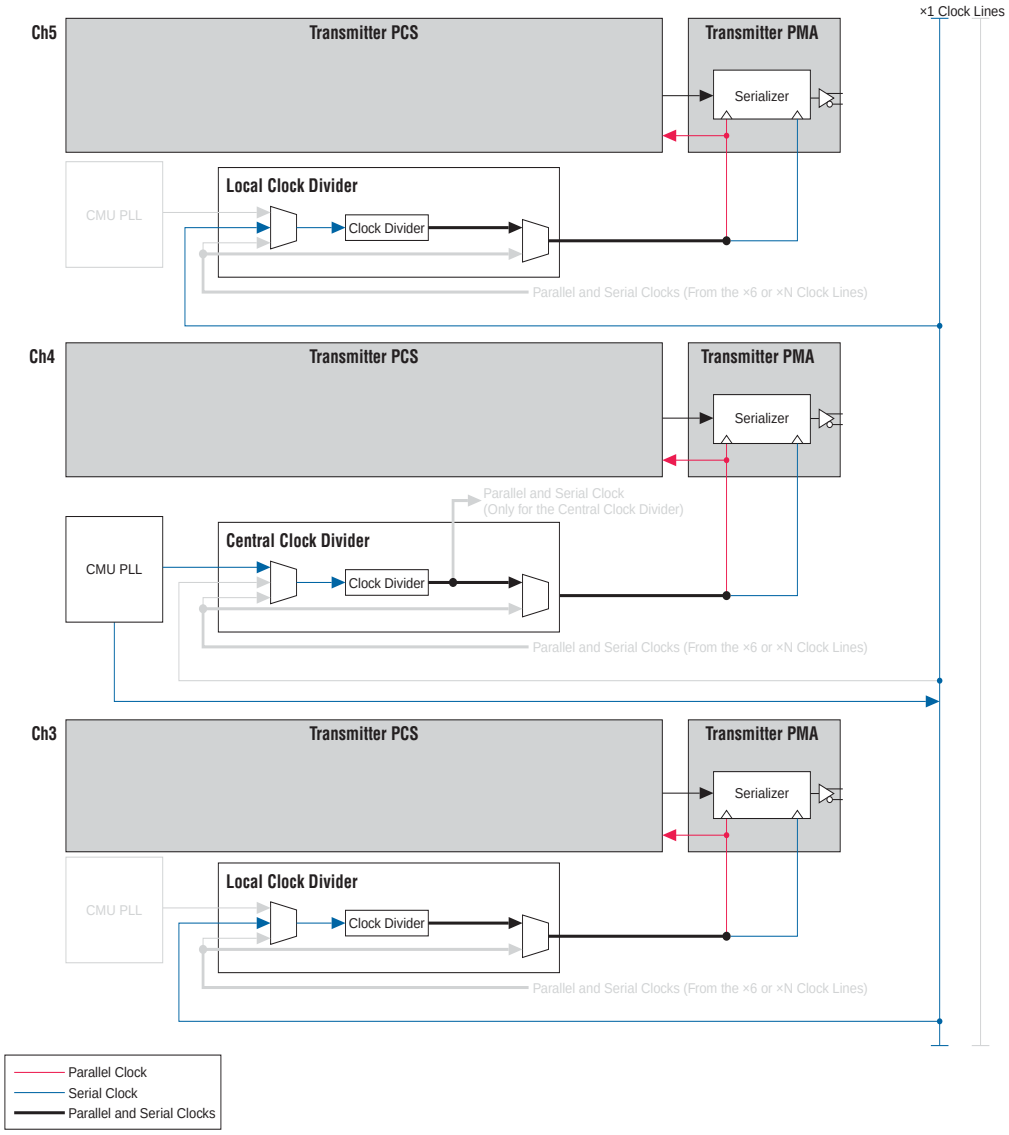
In the standard PCS channel, the parallel clock is used by all the blocks up to the read side of the TX phase compensation FIFO in all configurations that do not use the byte serializer block. For configurations that use the byte serializer block, the clock is divided by a factor of two for the byte serializer and the read side of the TX phase compensation FIFO. The clock used to clock the read side of the TX phase compensation FIFO is also forwarded to the FPGA fabric to provide an interface between the FPGA fabric and the transceiver.

For more information about clocking schemes used in different configurations, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

Non-Bonded Channel Configurations

In non-bonded channel configurations, the parallel clock is generated by the clock divider of individual channels. Figure 2–14 shows three transmitter-only channels in non-bonded configuration driven by the channel PLL of channel 4 configured as a CMU PLL driving the x1 clock line. The clock divider block of each channel generates its own parallel clock by dividing the serial clock from the x1 clock line.

Figure 2–14. Three Transmitter-Only Channels Configured in Non-Bonded Configuration

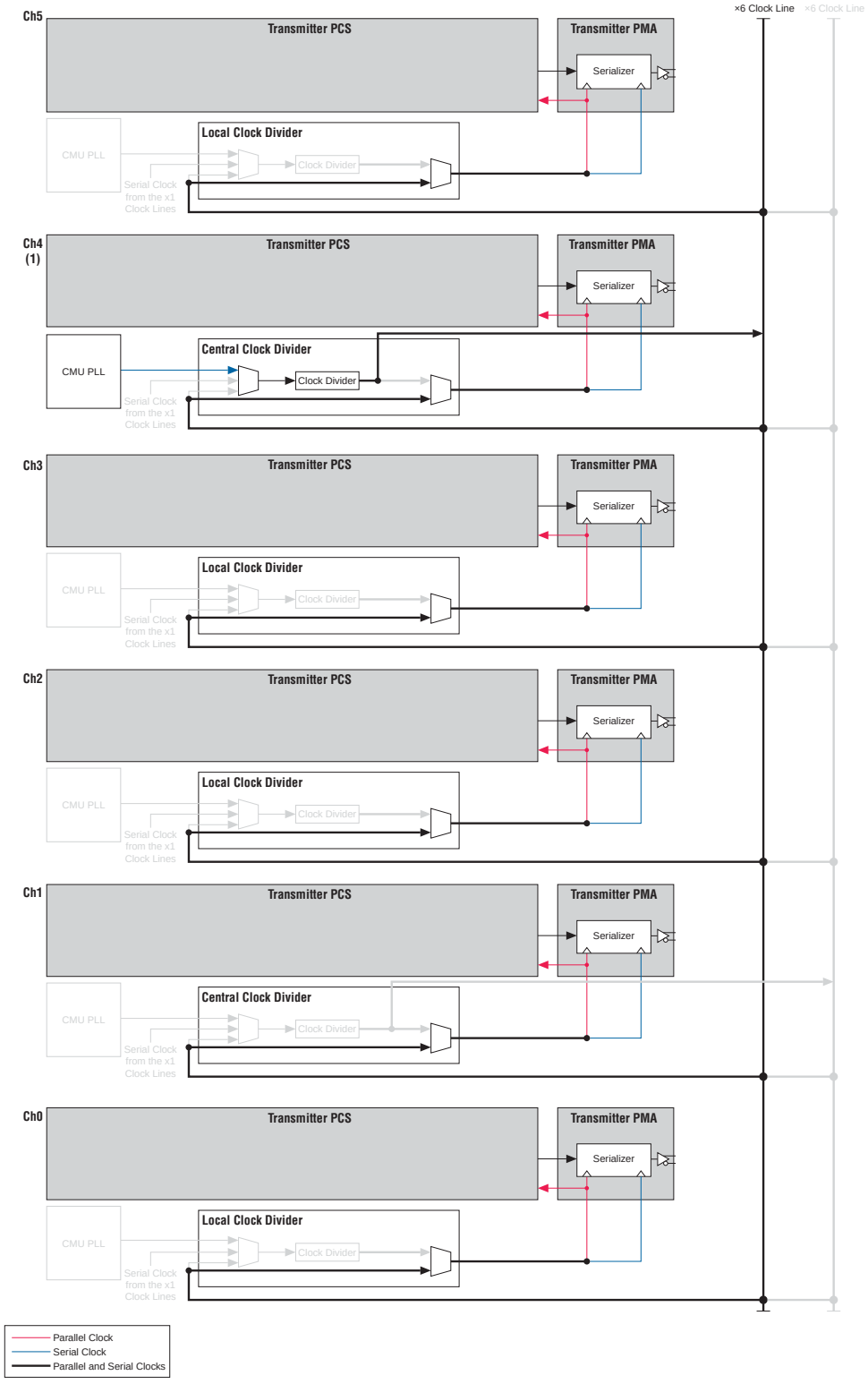


Bonded Channel Configurations

In bonded configurations, both the parallel clock and serial clock are sourced from either the x6 or xN clock line. The central clock dividers source the serial clock from a transmitter PLL from the same transceiver bank using the x1 clock line. The central clock divider generates the parallel clock and drives both the serial clock and parallel clock on the x6 clock line, which can drive the xN clock line.

Figure 2–15 shows five transmitter-only channels configured in bonded configuration and driven by the channel PLL of channel 4 configured as a CMU PLL. The central clock divider of channel 4 generates a parallel clock and drives both the serial clock and parallel clock on the x6 clock line. All bonded channels source both serial and parallel clocks from the x6 clock line.

Figure 2–15. Five Transmitter-Only Channels Configured in Bonded Configuration



Note to Figure 2–15:

(1) Channel 4 cannot be used as transmitter channel since the CMU PLL is used to generate clocks for other channels.

For an example of using the xN clock lines, refer to the PCIe x8 configuration in the "PCI Express (PCIe)—Gen1, Gen2, and Gen3" section of the *Transceiver Configurations in Stratix V Devices* chapter.

Bonded Channel Configurations Using the PLL Feedback Compensation Path

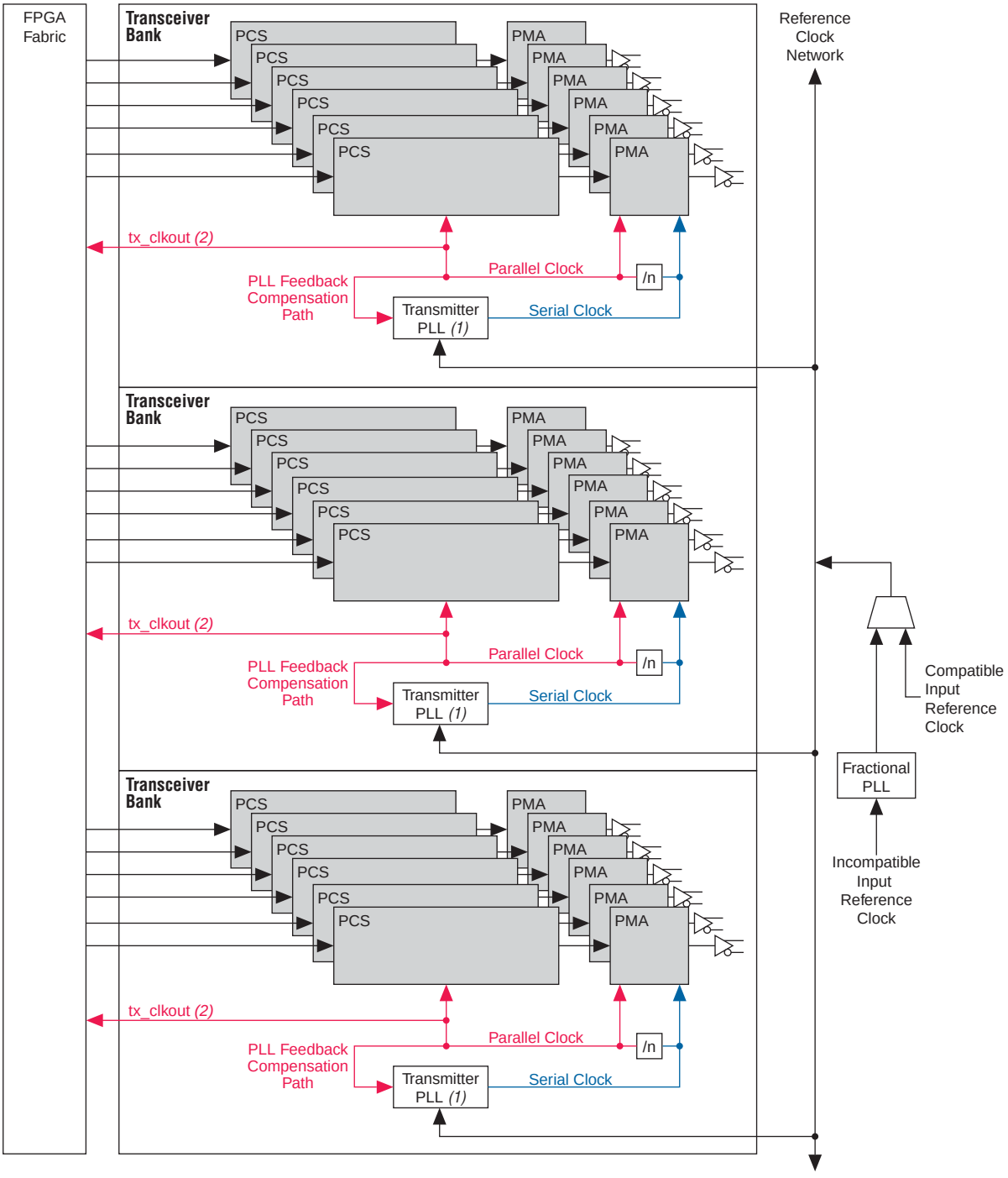
You can bond channels across multiple banks by using the PLL feedback compensation path. The PLL feedback compensation path loops the parallel clock, which is used by the PCS blocks, back to the transmitter PLL. The PLL feedback compensation path synchronizes the parallel clock used to clock the PCS blocks in all transceiver banks with the `refclk`. You can use the PLL feedback compensation path to reduce channel-to-channel skew, which is introduced by the clock divider in each transceiver bank.

To bond channels using the PLL feedback compensation path, the input reference clock frequency used by the transmitter PLL must be the same as the parallel clock that clocks the PCS of the same channel.

If the input reference clock frequency is not equal to the parallel clock frequency, use a fractional PLL to synthesize an input reference clock with the same frequency as the parallel clock.

Figure 2–16 shows 18 transmitter channels within three transceiver banks that are bonded using the PLL feedback compensation path. A fractional PLL is used to synthesize a clock that has the same frequency as the parallel clocks, which are looped back to the transmitter PLL.

Figure 2–16. Three Transceiver Bank Channels Bonded Using the PLL Feedback Compensation Path



Notes to Figure 2–16:

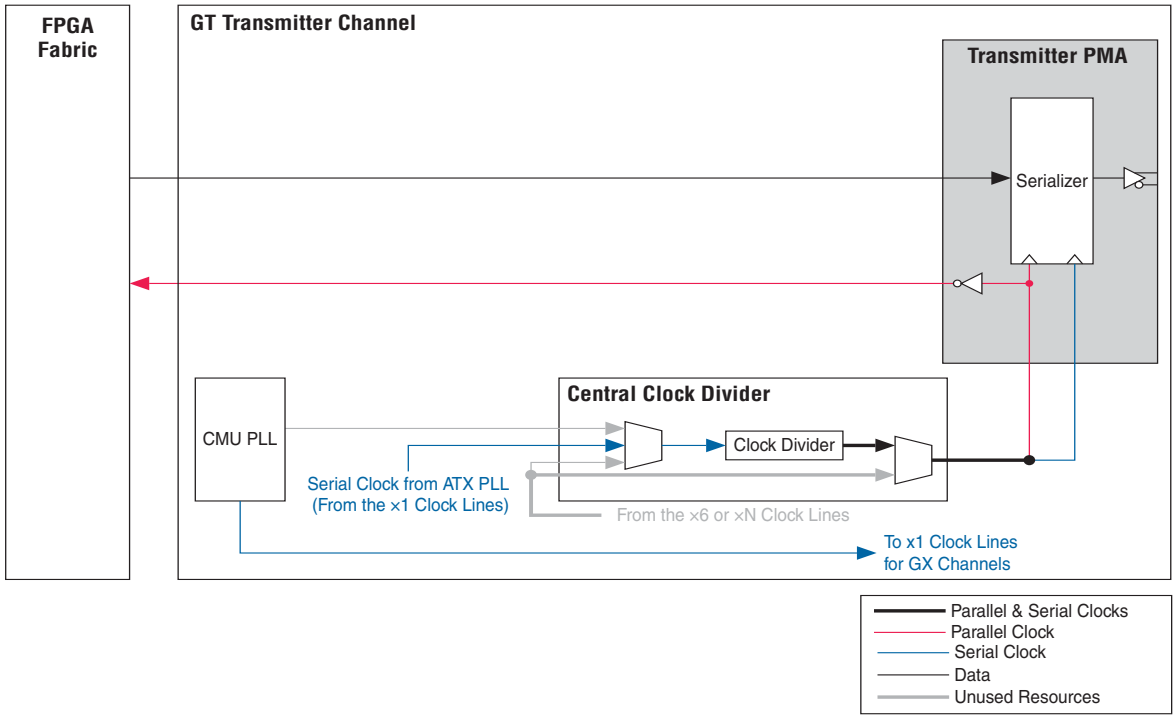
(1) The transmitter PLL can be an ATX PLL or a CMU PLL. You can have up to six channels per bank with an ATX PLL and five channels per bank with a CMU PLL.

(2) `tx_clkout` from any of the banks can be used with the FPGA fabric-transceiver interface for all the bonded channels.

Transmitter GT Channel Clocking

Figure 2–17 shows the clocking for the GT transmitter channel.

Figure 2–17. GT Channel Transmitter Clocking



As shown in Figure 2–17, the central clock divider block provides the serial and the parallel clocks to the serializer.

In a GT transmitter channel, the parallel clock is forwarded to the FPGA fabric to provide an interface between the FPGA fabric and the transceiver. All PCS functions, such as encoding and bit slipping, must be implemented in the FPGA core.

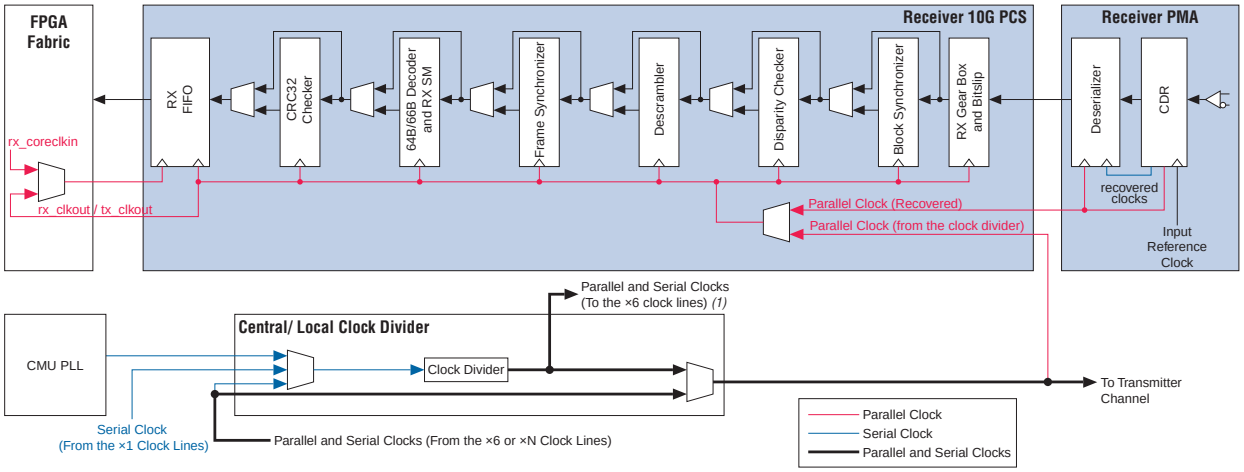
Receiver Clocking

Receiver clocking refers to the internal clocking architecture of the receiver channel of a transceiver. This section discusses receiver clocking for GX and GT receiver channels.

GX Channel Receiver Clocking

Figure 2–18 shows clocking for the receiver 10G PCS and the receiver PMA.

Figure 2–18. Receiver 10G PCS Clocking

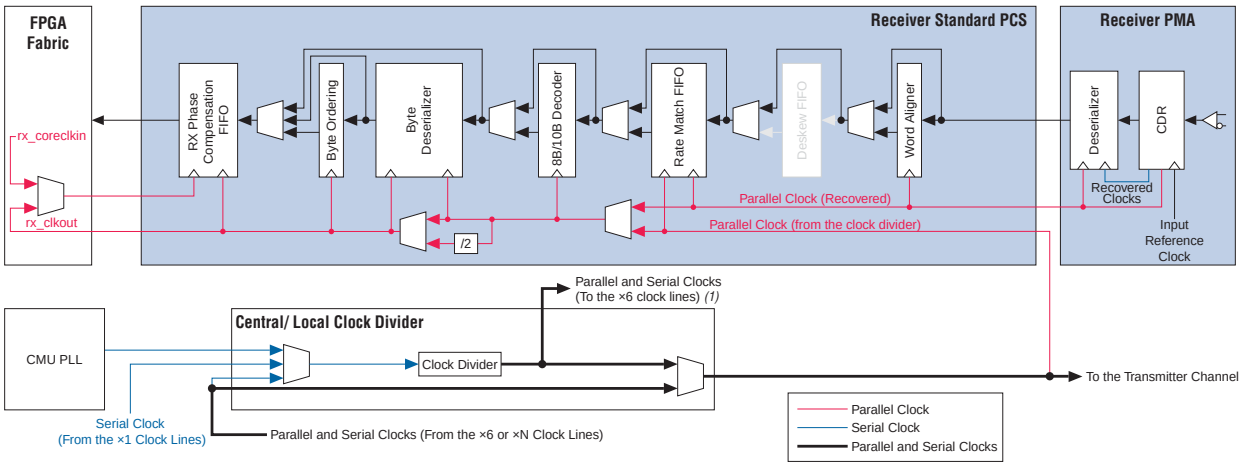


Note to Figure 2–18:

(1) Available only in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

Figure 2–19 shows clocking for the receiver standard PCS and receiver PMA.

Figure 2–19. Receiver Standard PCS Clocking



Note to Figure 2–19:

(1) Available only in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

The CDR in the PMA of each channel recovers the serial clock from the incoming data. The CDR also divides the serial clock (recovered) to generate the parallel clock (recovered). Both clocks are used by the deserializer. The receiver PCS can use the following clocks, depending on the configuration of the receiver channel:

- Parallel clock (recovered) from the CDR in the PMA
- Parallel clock from the clock divider used by the transmitter PCS for that channel

Table 2–2 lists the clock sources that are available for each block in the receiver PCS.

Table 2–2. Clock Sources for All Receiver PCS Blocks

PCS	Block	Clock Source
Standard	Word aligner	Parallel clock (recovered)
	Rate match FIFO	Write side: parallel clock (recovered) Read side: parallel clock from the clock divider
	8B/10B decoder	If rate matcher is not used: parallel clock (recovered) If rate matcher is used: parallel clock from the clock divider
	Byte deserializer	Write side: ■ If rate matcher is not used: parallel clock (recovered) ■ If rate matcher is used: parallel clock from the clock divider Read side: Divided down version of the write side clock, depending on the deserialization factor of 1 or 2, also called the parallel clock (divided)
	Byte ordering	Parallel clock (divided)
	Receiver (RX) phase compensation FIFO	Write Side: Parallel clock (divided). This clock is also forwarded to the FPGA fabric Read Side: Clock sourced from the FPGA fabric
10G	All PCS blocks	Regular mode: parallel clock (recovered) Loopback mode: parallel clock from the clock divider *(1)*

Note to Table 2–2:

(1) For more information about loopback mode, refer to the *Transceiver Loopback Support in Stratix V Devices* chapter.

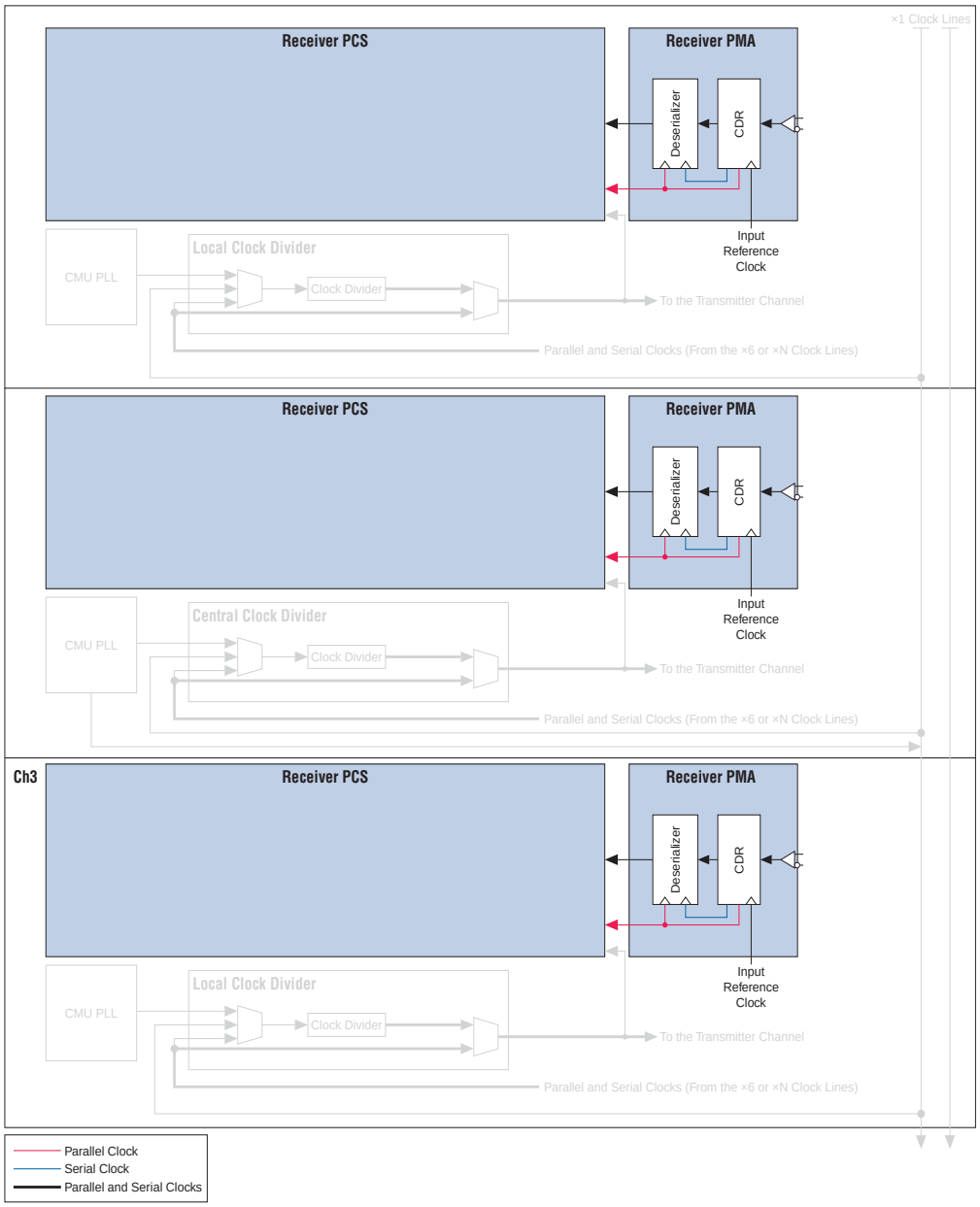
Non-Bonded Channel Configurations

In non-bonded configurations, the receiver standard PCS requires both the parallel clock (recovered) and parallel clock from the clock divider. Depending on the configuration, the receiver PCS may require the parallel clock from the clock divider that is used for the transmitter PCS.

In non-bonded configurations, the receiver 10G PCS uses only the parallel clock (recovered) for all its blocks.

Figure 2–20 shows three channels configured in non-bonded configuration. They use the receiver standard PCS that does not use a rate match FIFO. The CDR of each channel recovers the serial clock (recovered) from the incoming data and generates the parallel clock (recovered) by dividing the serial clock (recovered). Depending on the configuration, the receiver PCS may also use the parallel clock from the clock divider that is generated by the local clock divider for the transmitter.

Figure 2–20. Three Channels Configured in Non-Bonded Configuration



For more information about the clocking scheme used in different configurations, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

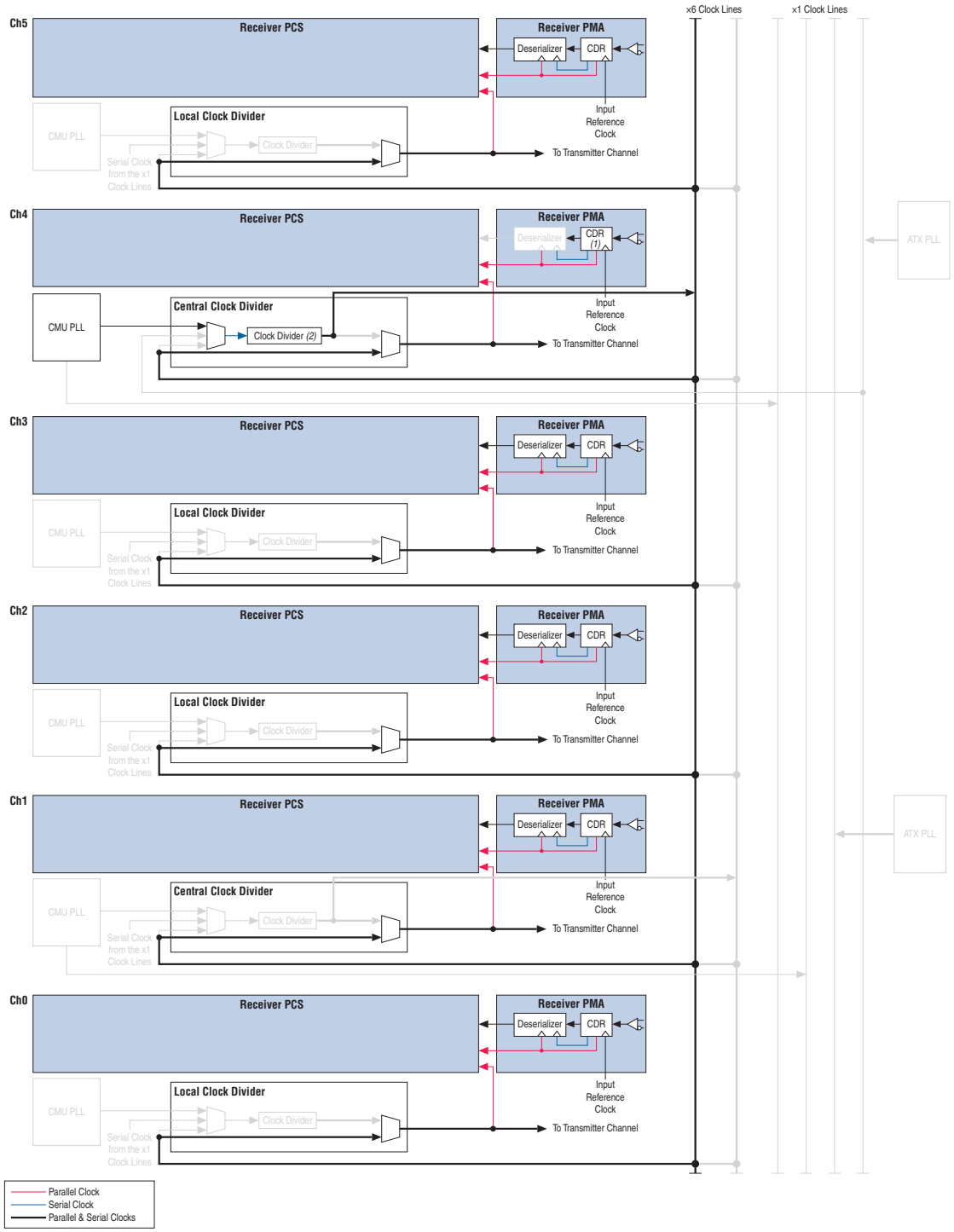
Bonded Channel Configurations

In bonded configurations, the receiver standard PCS requires both the parallel clock (recovered) and parallel clock from the clock divider.

In bonded configurations, the receiver 10G PCS uses only the parallel clock (recovered) for all its blocks.

Figure 2–21 shows five channels in a transceiver bank configured in bonded configuration, using the receiver standard PCS. The receiver PCS uses both the parallel clock (recovered) and parallel clock from the clock divider. The parallel clock from the clock divider is generated by the central clock divider for the transmitter PCS. It also drives some blocks in the receiver PCS, depending on the configuration you use.

Figure 2–21. Five Receiver Channels Configured in Bonded Duplex Configuration



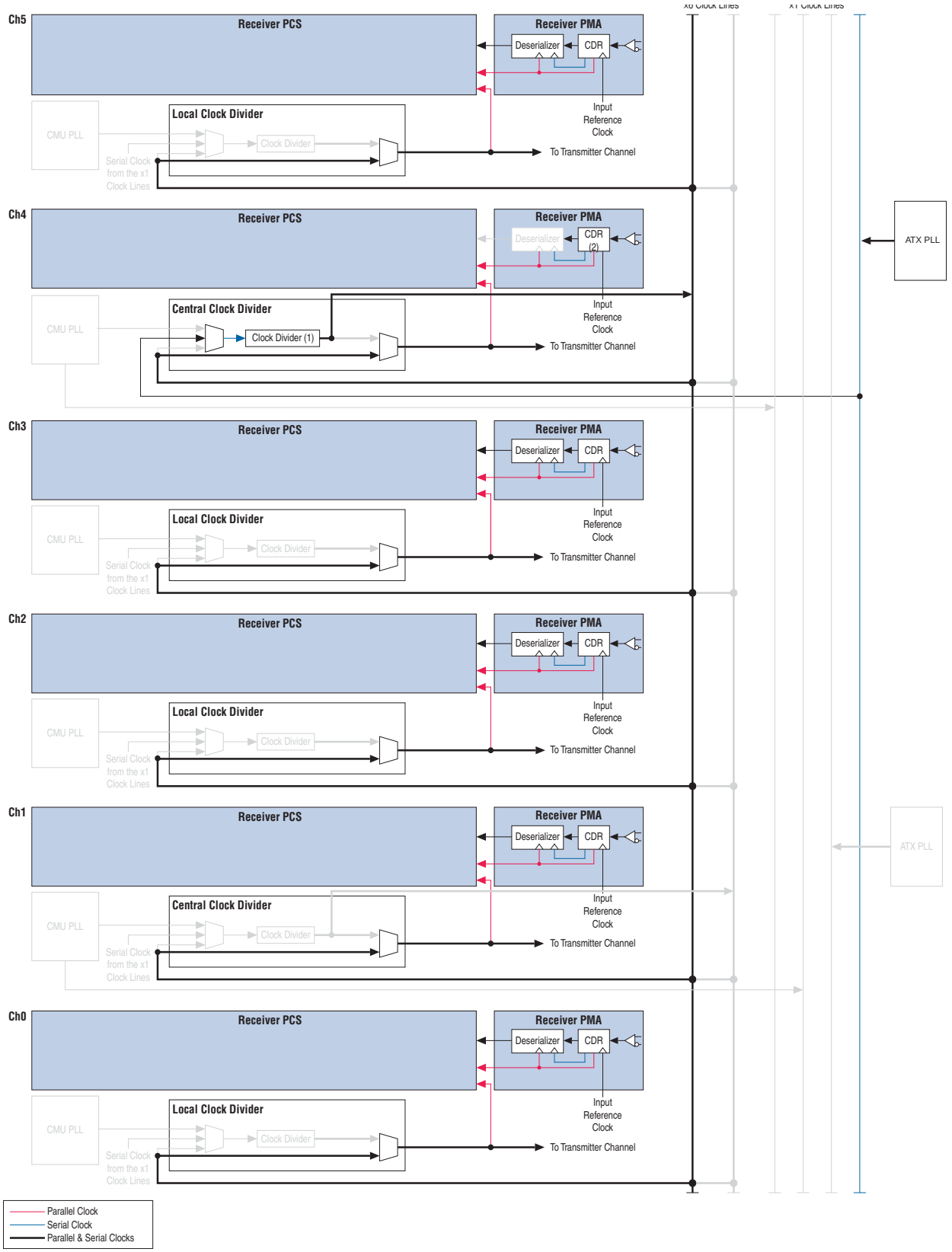
Notes to Figure 2–21:

(1) You cannot use channel 4 as a receiver as the channel PLL is being used as a CMU PLL rather than a CDR.

(2) You cannot use channel 4 as a transmitter as it is required for bonding non-contiguous channel 3 and channel 5.

Figure 2–22 shows all six channels in the transceiver bank in bonded configuration, as opposed to a maximum of five, as shown in Figure 2–21. Six channel bonding is possible because the ATX PLL is used as a transmitter PLL instead of a channel PLL in the transceiver bank. Using the ATX PLL allows you to use the channel PLLs of both channels 1 and 4 as CDRs to perform receiver operations.

Figure 2–22. Six Channels Configured in Bonded Configuration Using ATX PLL

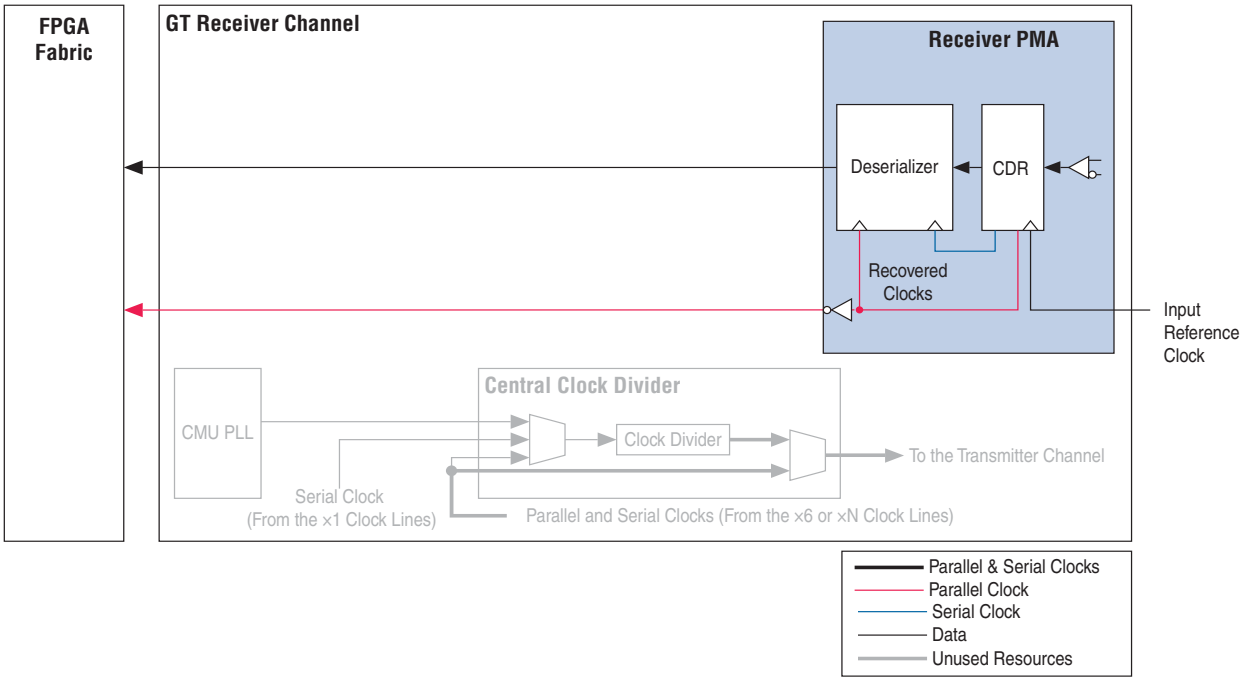


For more information about the clocking scheme used in different configurations, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

GT Channel Receiver Clocking

Figure 2–23 shows clocking for the GT channel receiver clocking.

Figure 2–23. GT Channel Receiver Clocking



The CDR in the PMA of the GT receiver channel recovers the serial clock from the incoming data and is driven by an input reference clock or clock from the reference clock network in the same GT transceiver bank. The CDR also divides the serial clock (recovered) to generate the parallel clock (recovered). Both clocks are used by the deserializer. The parallel clock (recovered) is forwarded to the FPGA fabric to interface the FPGA fabric with the transceiver. All PCS functions, such as word alignment, rate matching, decoding, and byte ordering, must be implemented in the FPGA core because the PCS is unavailable in the GT receiver channel.

FPGA Fabric-Transceiver Interface Clocking

The FPGA fabric-transceiver interface clocks consist of clock signals from the FPGA fabric to the transceiver blocks and clock signals from the transceiver blocks to the FPGA fabric. These clock resources use the clock networks in the FPGA core, including the global (GCLK), regional (RCLK), and periphery (PCLK) clock networks.

The FPGA fabric-transceiver interface clocks can be subdivided into the following three categories:

- Input reference clocks—Refer to "Input Reference Clock Sources" on page 2–2. The input reference clock can be an FPGA fabric-transceiver interface clock when it is also forwarded to the FPGA fabric to clock the logic in the FPGA fabric.

- Transceiver datapath interface clocks—Used to transfer data, control, and status signals between the FPGA fabric and the transceiver channels. The transceiver channel forwards the tx_clkout signal to the FPGA fabric to clock the data and control signals into the transmitter. The transceiver channel also forwards the recovered rx_clkout clock (in configurations without the rate matcher) or the tx_clkout clock (in configurations with the rate matcher) to the FPGA fabric to clock the data and status signals from the receiver into the FPGA fabric.
- Other transceiver clocks—Used to form a part of the FPGA fabric-transceiver interface clocks as follows:
  - mgmt_clk—Avalon®-MM interface clock used for controlling the transceivers, dynamic reconfiguration, and calibration
  - fixed_clk—125 MHz fixed-rate clock used in the PCIe (PIPE) receiver detect circuitry

Table 2–3 lists the FPGA fabric-transceiver interface clocks.

Table 2–3. FPGA Fabric-Transceiver Interface Clocks *(1)*

Clock Name	Clock Description	Interface Direction	FPGA Fabric Clock Resource Utilization
pll_ref_clk	Input reference clock used for clocking logic in the FPGA fabric.	Transceiver-to-FPGA fabric	GCLK, RCLK, PCLK
tx_clkout	Clock forwarded by the transceiver for clocking the transceiver datapath interface. The value of tx_clkout is derived by dividing the data rate by the serialization factor. For example, a 3 Gbps link with a serialization factor of 20 will result in a tx_clkout of 150 MHz.		
rx_clkout	Clock forwarded by the receiver for clocking the receiver datapath interface. The value of rx_clkout is derived by dividing the data rate by the deserialization factor. For example, a 10 Gbps link with a deserialization factor of 40 will result in a rx_clkout of 250 MHz.		
tx_coreclkin	User-selected clock for clocking the transmitter datapath interface.	FPGA fabric-to-transceiver	
rx_coreclkin	User-selected clock for clocking the receiver datapath interface.		
fixed_clk	PCIe receiver detect clock.		
mgmt_clk *(2)*	Avalon-MM interface management clock.		

Notes to Table 2–3:

(1) For more information about the GCLK, RCLK, and PCLK resources available in each device, refer to the *Clock Networks and PLLs in Stratix V Devices* chapter.

(2) The mgmt_clk is a free-running clock that is not derived from the transceiver blocks, except if mgmt_clk is derived from the dedicated refclk pin.

You can forward the pll_ref_clk, tx_clkout, and rx_clkout clocks to a fractional PLL so that the fractional PLL can synthesize a clock for the FPGA logic.

Table 2–4 lists the port names for `tx_clkout` and `rx_clkout`.

Table 2–4. Configuration Specific Port Names for tx_clkout and rx_clkout

Configuration	Port Name for tx_clkout	Port Name for rx_clkout
Custom	`tx_clkout`	`rx_clkout`
Interlaken	`tx_clkout`	`rx_clkout`
Low Latency	`tx_clkout`	`rx_clkout`
PCIe	`pipe_pclk`	`pipe_pclk`
XAUI	`xgmii_tx_clk`	`xgmii_rx_clk`

Transmitter Datapath Interface Clocking

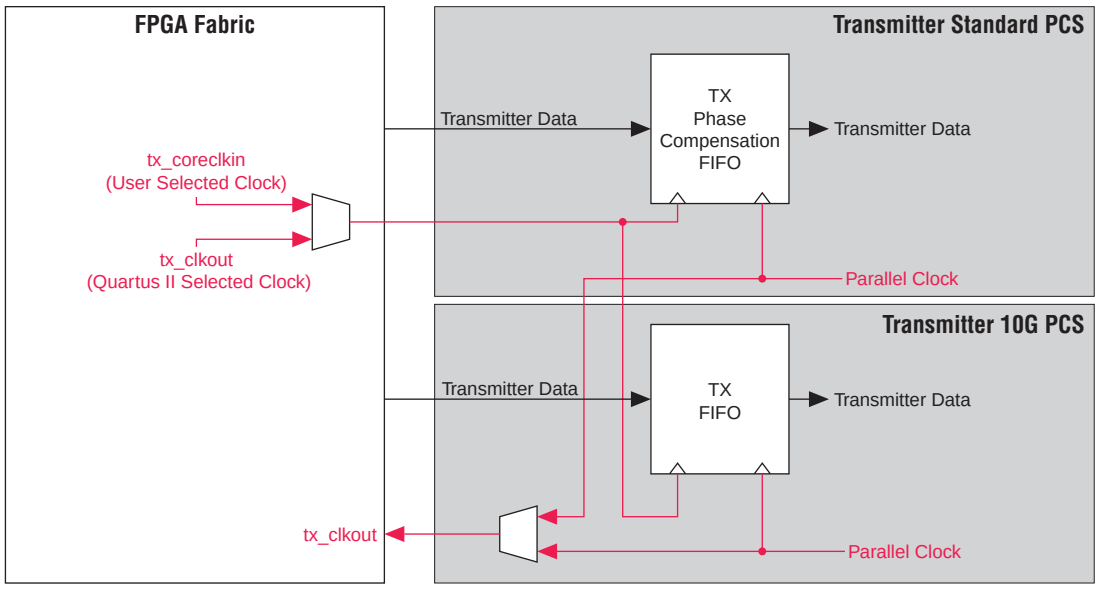
The transmitter datapath interface is comprised of the following:

- Write side of the TX phase compensation FIFO—for configurations that use the standard PCS channel
- Write side of the TX FIFO—for configurations that use the 10G PCS channel

This interface is clocked by the transmitter datapath interface clock. Figure 2–24 shows transmitter datapath interface clocking. The transmitter PCS forwards the following clocks to the FPGA fabric:

- `tx_clkout` for each transmitter channel in non-bonded configuration
- `tx_clkout[0]` for all transmitter channels in bonded configuration

Figure 2–24. Transmitter Datapath Interface Clocking



All configurations using the standard PCS channel must have a 0 parts per million (ppm) difference between the transmitter datapath interface clock and the read side clock of the TX phase compensation FIFO.

For more information about interface clocking for each configuration, refer to the *Transceiver Configurations in Stratix V Devices* chapter.

You can clock the transmitter datapath interface by using one of the following:

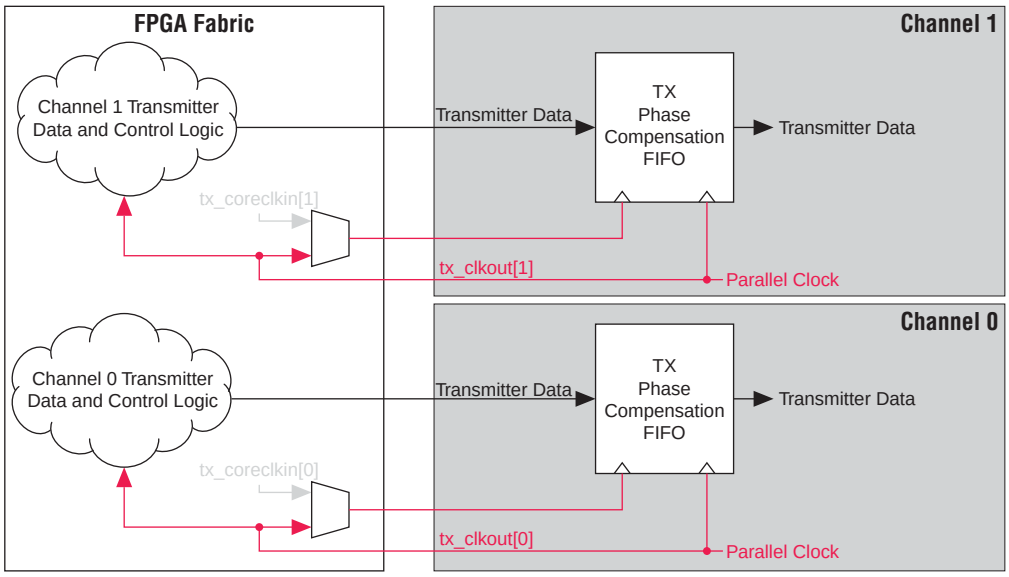
- Quartus II-selected transmitter datapath interface clock
- User-selected transmitter datapath interface clock

User selection allows you to share the transceiver datapath interface clocks to reduce GCLK, RCLK, and PCLK resource utilization in your design.

Quartus II-Selected Transmitter Datapath Interface Clock

The Quartus II software automatically picks the appropriate clock from the FPGA fabric to clock the transmitter datapath interface. Figure 2–25 shows the transmitter datapath interface of two non-bonded channels clocked by their respective transmitter PCS clocks that are forwarded to the FPGA fabric.

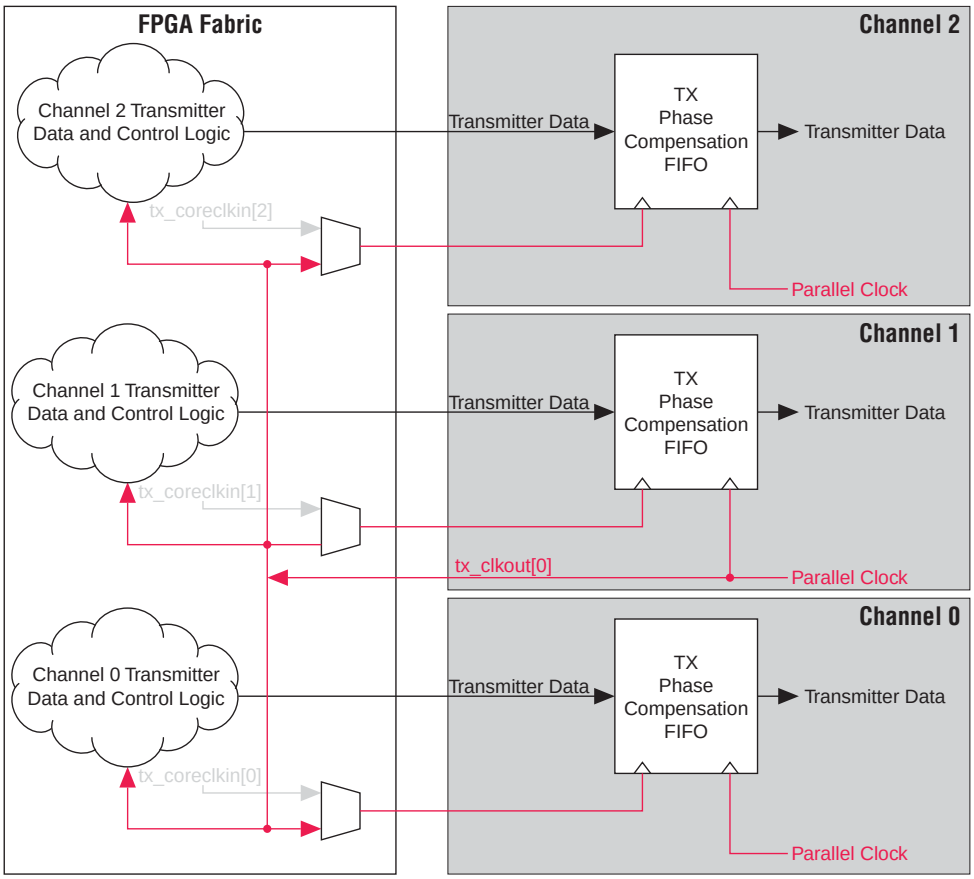
Figure 2–25. Transmitter Datapath Interface Clocking for Non-Bonded Channels



The FPGA fabric-transceiver interface clocking for GT transmitter channels is similar to the non-bonded GX transmitter channel FPGA fabric-transceiver interface clocking.

Figure 2–26 shows the transmitter datapath interface of three bonded channels clocked by the `tx_clkout[0]` clock. The `tx_clkout[0]` clock is derived from the central clock divider of channel 1 or 4 in a transceiver bank.

Figure 2–26. Transmitter Datapath Interface Clocking for Three Bonded Channels



User-Selected Transmitter Datapath Interface Clock

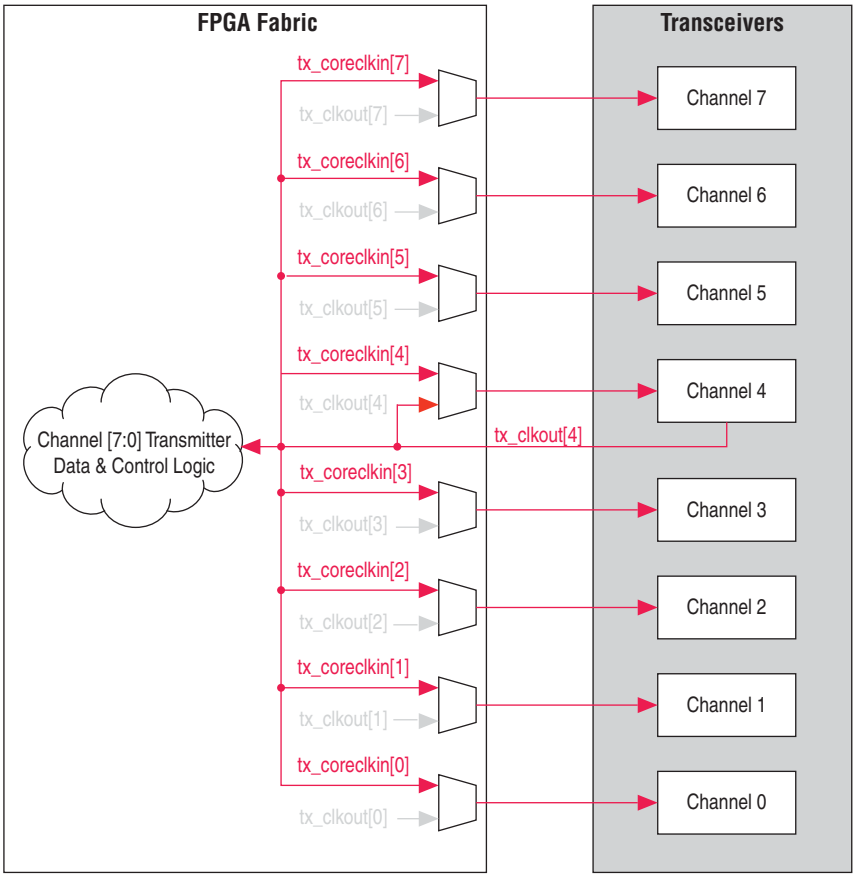
Multiple transmitter channels that are non-bonded lead to high utilization of GCLK, RCLK, and PCLK resources (one clock resource per channel as shown in Figure 2–25). You can significantly reduce GCLK, RCLK, and PCLK resource use for transmitter datapath clocks if the transmitter channels are identical.

Identical transmitter channels are defined as channels that have the same input reference clock source, the same transmitter PLL configuration, and the same transmitter PMA and PCS configuration. Identical transmitter channels may have different analog settings, such as transmitter voltage output differential (V_{OD}), transmitter common mode voltage (V_{CM}), or pre-emphasis.

To achieve the clock resource savings, select a common clock driver for the transmitter datapath interface of all identical transmitter channels. Figure 2–27 shows eight identical channels clocked by a single clock (`tx_clkout` of channel 4). To clock eight identical channels with a single clock, instantiate the `tx_coreclkin` port for all the identical transmitter channels (`tx_coreclkin[7:0]`). Connect `tx_clkout[4]` to the `tx_coreclkin[7:0]` ports. Also, connect `tx_clkout[4]` to the transmitter data and control logic for all eight channels.

Resetting or powering down channel 4 will lead to a loss of the clock for all eight channels.

Figure 2–27. Eight Identical Channels with a Single User-Selected Transmitter Interface Clock



The common clock must have a 0 ppm difference with respect to the read side of the TX FIFO (in the 10G PCS channel) or TX phase compensation FIFO (in the standard PCS channel) for all the identical channels. A frequency difference causes the FIFO to underflow or overflow, depending on whether the common clock is slower or faster, respectively. You can drive the "GXB 0 PPM Core Clock Assignment" by one of the following sources:

- `tx_clkout` of any channel in non-bonded channel configurations
- `tx_clkout[0]` in bonded channel configurations
- When there is 0 PPM between `refclk` and `tx_clkout`

The Quartus II software does not allow gated clocks or clocks generated in the FPGA logic to drive the `tx_coreclkin` ports.

Because the Quartus II software allows you to use external pins, such as dedicated `refclk` pins, it has no way of ensuring a 0 ppm difference. You must ensure a 0 ppm difference. For more information about the 0 PPM setting, refer to "GXB 0 PPM Core Clock Assignment" on page 2–39".

Receiver Datapath Interface Clock

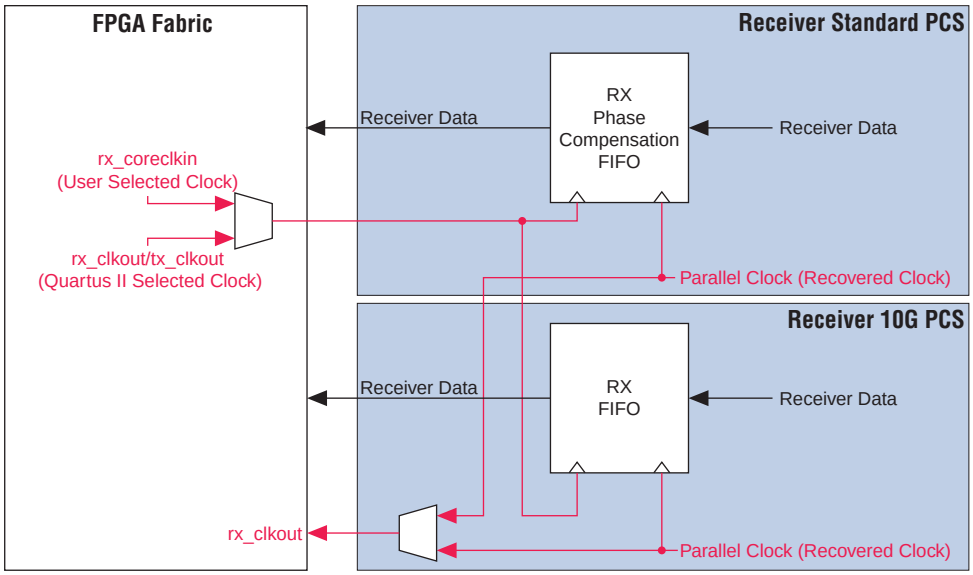
The receiver datapath interface is comprised of the following:

- Read side of the RX phase compensation FIFO—for configurations that use the standard PCS channel
- Read side of the RX FIFO—for configurations that use the 10G PCS channel

This interface is clocked by the receiver datapath interface clock. Figure 2–28 shows receiver datapath interface clocking. The receiver PCS forwards the following clocks to the FPGA fabric:

- `rx_clkout`—for each receiver channel in non-bonded configuration when a rate matcher is not used
- `tx_clkout`—for each receiver channel in non-bonded configuration when a rate matcher is used
- single `tx_clkout[0]`—for all receiver channels in bonded configuration

Figure 2–28. Receiver Datapath Interface Clocking



All configurations that use the standard PCS channel must have a 0 ppm difference between the receiver datapath interface clock and the read side clock of the RX phase compensation FIFO.

For more information about interface clocking for each configuration, refer to the clocking sections for each configuration in the *Transceiver Configurations in Stratix V Devices* chapter.

You can clock the receiver datapath interface by using one of the following:

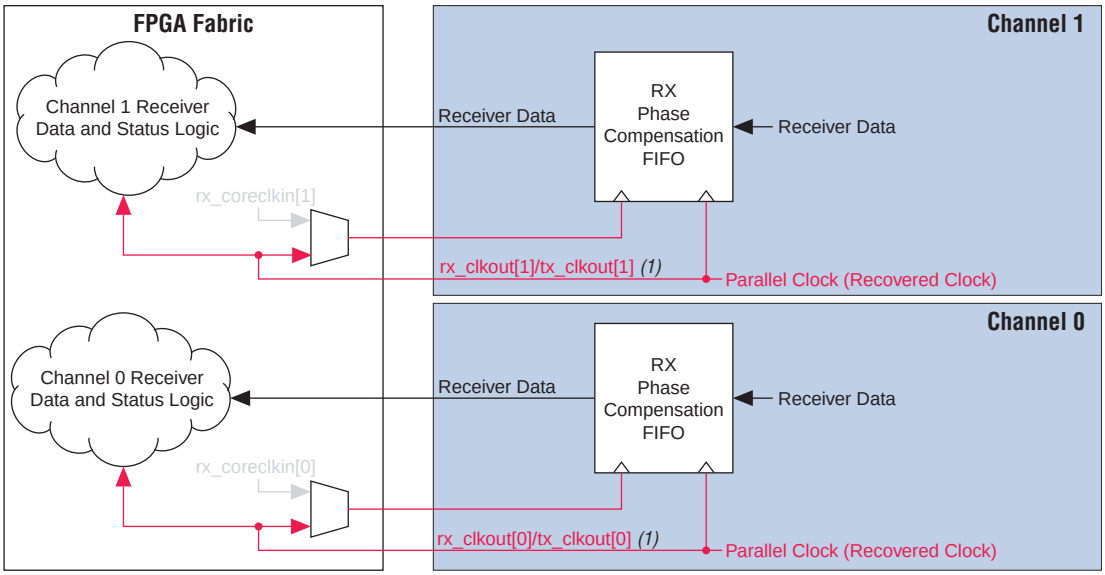
- Quartus II-selected receiver datapath interface clock
- User-selected receiver datapath interface clock

User-selection is provided to share the transceiver datapath interface clocks to reduce GCLK, RCLK, and PCLK resource utilization in your design.

Quartus II Software-Selected Receiver Datapath Interface Clock

The Quartus II software automatically picks the appropriate clock from the FPGA fabric to clock the receiver datapath interface. Figure 2–29 shows the receiver datapath interface of two non-bonded channels that are clocked by their respective receiver PCS clocks and forwarded to the FPGA fabric.

Figure 2–29. Receiver Datapath Interface Clocking for Non-Bonded Channels



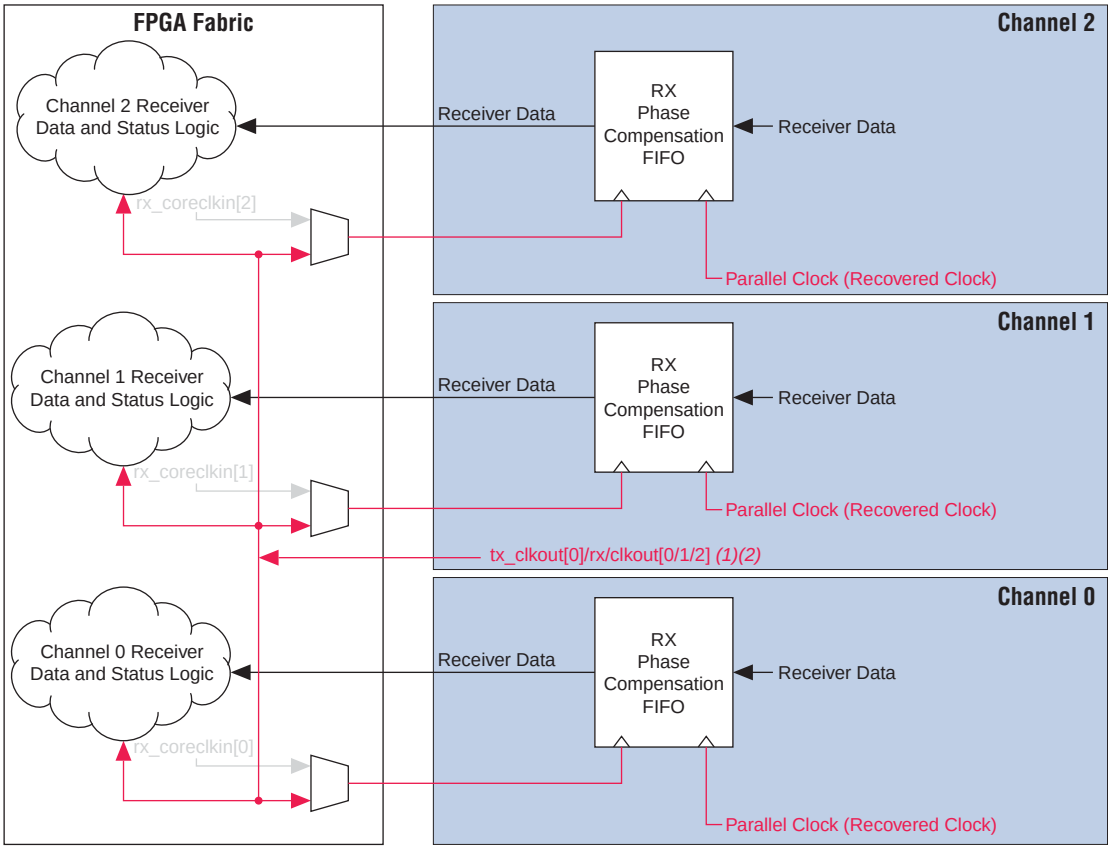
Note to Figure 2–29:

(1) `tx_clkout` can only be used if a rate matcher is used.

The FPGA fabric-transceiver interface clocking for GT receiver channels is similar to the non-bonded GX receiver channel FPGA fabric-transceiver interface clocking.

Figure 2–30 shows the receiver datapath interface of three bonded channels clocked by the tx_clkout[0] clock. The tx_clkout[0] clock is derived from the central clock divider of channel 1 or 4 in a transceiver bank.

Figure 2–30. Receiver Datapath Interface Clocking for Three Bonded Channels



Notes to Figure 2–30:

(1) tx_clkout[0] can only be used if a rate matcher is used.

(2) rx_clkout[0], rx_clkout[1] or rx_clkout[2] can be used to clock the read side of the rx phase compensation FIFO of all channels only if there is 0 ppm difference between the rx_clkout[0], rx_clkout[1], and rx_clkout[2].

User-Selected Receiver Datapath Interface Clock

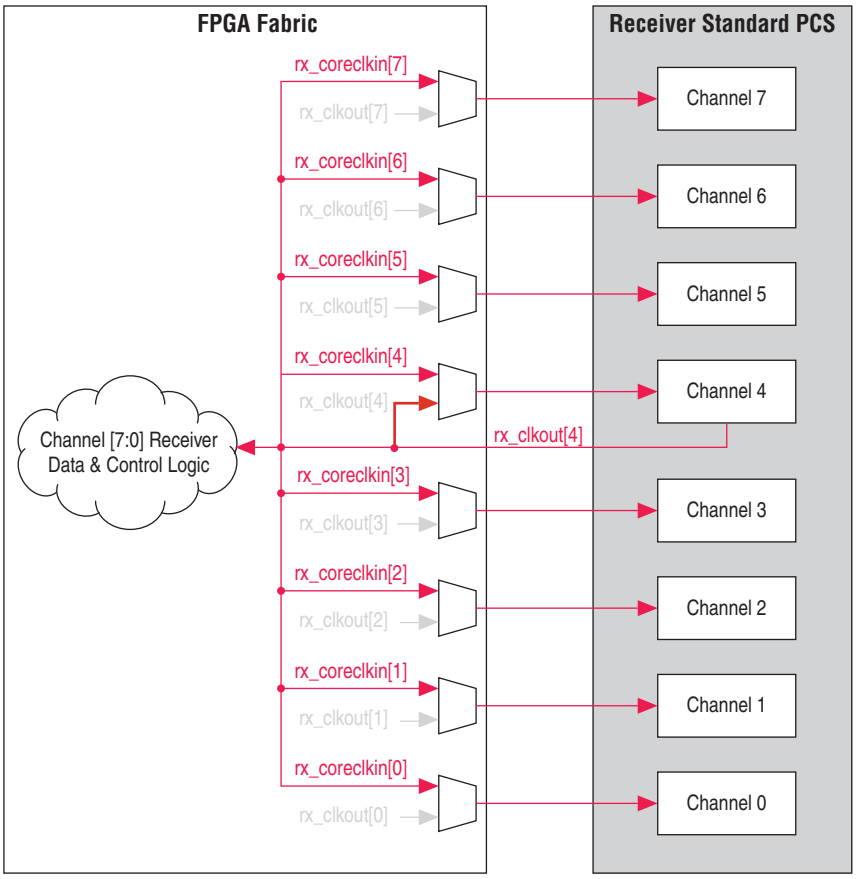
Multiple receiver channels that are non-bonded lead to high utilization of GCLK, RCLK, and PCLK resources (one clock resource per channel as shown in Figure 2–29 on page 2–36). You can significantly reduce GCLK, RCLK, and PCLK resource use for receiver datapath clocks if the receiver channels are identical.

Identical receiver channels are defined as channels that have the same input reference clock source for the CDR, and the same receiver PMA and PCS configuration. Identical receiver channels may have different analog settings, such as receiver common mode voltage (V_{ICM}), equalization, or DC gain.

To achieve clock resource savings, select a common clock driver for the receiver datapath interface of all identical receiver channels. To do this, instantiate the `rx_coreclkin` port for all the identical receiver channels and connect the common clock driver to the receiver datapath interface, and receiver data and control logic. Figure 2–31 shows eight identical channels that are clocked by a single clock (`rx_clkout` of channel 4). To clock eight identical channels with a single clock, instantiate the `rx_coreclkin` port for all the identical receiver channels (`rx_coreclkin[7:0]`). Connect `rx_clkout[4]` to the `rx_coreclkin[7:0]` ports. Also, connect `rx_clkout[4]` to the receiver data and control logic for all eight channels.

Resetting or powering down channel 4 leads to a loss of the clock for all eight channels.

Figure 2–31. Eight Identical Channels with a Single User-Selected Receiver Interface Clock



The common clock must have a 0 ppm difference with respect to the write side of the RX FIFO (in the 10G PCS channel) or RX phase compensation FIFO (in the standard PCS channel) for all the identical channels. A frequency difference causes the FIFO to underflow or overflow, depending on whether the common clock is faster or slower, respectively. You can drive the 0 ppm common clock driver with one of the following sources:

- `rx_clkout` of any channel in non-bonded receiver channel configurations without the rate matcher
- `tx_clkout` of any channel in non-bonded receiver channel configurations with the rate matcher
- `tx_clkout[0]` in bonded receiver channel configurations
- Dedicated `refclk` pins

The Quartus II software does not allow gated clocks or clocks generated in the FPGA logic to drive the `rx_coreclkin` ports.

Because the Quartus II software allows you to use external pins, such as dedicated `refclk` pins, it has no way of ensuring a 0 ppm difference. You must ensure a 0 ppm difference.

GXB 0 PPM Core Clock Assignment

The common clock should have a 0 PPM difference with respect to the read side of the TX FIFO (in the 10G PCS channel) or TX phase compensation FIFO (in the Standard PCS channel) of all the identical channels. A frequency difference causes the FIFO to under-run or overflow, depending on whether the common clock is slower or faster, respectively. The 0 PPM common clock driver can be driven by one of the following sources:

- `tx_clkout` in non-bonded channel configurations
- `tx_clkout[0]` in bonded channel configurations
- `rx_clkout` in non-bonded channel configurations
- `refclk` when there is 0 PPM `difference` between `refclk` and `tx_clkout`

Table 2–5 lists the 0 PPM core clock settings that you make in the Quartus II Assignment Editor.

Table 2–5. 0 PPM Core Clock Settings

Assignments	Description
To	`tx_dataout/rx_datain` pins of all channels whose `tx/rx_coreclk` ports are connected together and driven by the 0 PPM clock driver.
Assignment Name	0 PPM coreclk setting
Value	ON

Note to Table 2–5:

(1) You can find the full hierarchy name of the 0 PPM clock driver using the **Node Finder** feature in the Quartus II Assignment Editor.

For more information about how 0 PPM is used with various transceiver PHYs, refer to the *Altera Transceiver PHY IP Core User Guide*.

Document Revision History

Table 2–6 lists the revision history for this document.

Table 2–6. Document Revision History

Date	Version	Changes
June 2012	1.6	■ Updated for the Quartus II software version 12.0 and reordered paragraphs. ■ Added Clock Divider section previously located in architecture chapter. ■ Added information about GXB 0 PPM core clock assignment. ■ Updated Figures 2-4, 2-7, 2-17, 2-21, 2-23, and 2-30.
February 2012	1.5	■ Updated document and figures for clarity. ■ Edited Figures 2-2, 2-3, and 2-4.
December 2011	1.4	■ Updated document and figures for clarity. ■ Changed path for serial and parallel clocks in channel 4 in Figure 2–14, Figure 2–20, and Figure 2–21.
November 2011	1.3	■ Added information about GT transceivers ■ Added information about bonding channels across transceiver banks by using the PLL feedback compensation path ■ x8 bonding using the xN clock lines is now available for PCIe Gen3 ■ Added information about the transceiver clocks used in the FPGA fabric ■ Added information about fractional mode when using fractional PLLs ■ Added information about using the FPGA fabric clocks as a reference clock to GX transceiver channels ■ Added information about forwarding transceiver clocks to a fractional PLL so that the fractional PLL can synthesize a clock for the FPGA logic
May 2011	1.2	■ Added information about fractional PLLs as they provide an input reference clock in “Input Reference Clocking.” ■ Chapter moved to Volume 3.
December 2010	1.1	■ Updated clock names ■ Updated figures for more accurate depiction of transceiver clocking ■ Added information about ATX PLLs
July 2010	1.0	Initial release

3. Transceiver Reset Control in Stratix V Devices

SV52004-3.3

This chapter provides the recommended transceiver initialization and reset sequence for Stratix® V devices. Altera's recommended reset sequence ensures that both the physical coding sublayer (PCS) and physical medium attachment (PMA) in each transceiver channel are initialized and functioning correctly.

Transceiver Reset Sequence

There are three methods of performing the reset sequence in Stratix V transceivers:

- "PHY IP Embedded Reset Controller"
  - The PHY IP embedded reset controller is enabled by default in all transceiver PHY IP cores except Native PHY IP core.
  - For bonded mode, the resets are shared between the bonded channels. Resetting one channel resets all the bonded channels. There is a single `tx_ready` and `rx_ready` status signal.
  - For non-bonded mode, there is a separate reset for all the receiver channels, but the transmitter channels share a common reset. Resetting one transmitter channel does not reset the other channels. There is a `single tx_ready` and `rx_ready` status signal.
  - The embedded reset controller is level sensitive (active high).
- "User-Controlled Reset Controller"
  - On non-protocol Transceiver PHY IP cores (Custom, Low Latency, and Deterministic Latency), you can disable the embedded reset controller and implement a user-controlled reset logic.
  - User-controlled reset controller is the default option for Native PHY IP Core.
  - The internal signals of the embedded reset controller are exposed as ports.
  - The user-controlled reset controller can be designed to reset a single channel or multiple channels simultaneously.
  - The user-controlled reset controller can be level sensitive (active high).

Altera also offers a Transceiver PHY reset controller that you can use instead of writing the complete user-controlled reset logic. The Transceiver PHY reset controller is available only for non-protocol Transceiver PHY IP cores (Custom, Low Latency, Deterministic Latency, and Native PHY). Refer to the "Transceiver PHY Reset Controller IP Core" section of the *Altera Transceiver PHY IP Core User Guide*.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

- Avalon® Memory-Mapped Reset Control and Status Registers

Transceiver Blocks Affected by the Reset and Powerdown Signals

Table 3–1 lists the blocks that are affected by specific reset and powerdown signals.

Table 3–1. Transceiver Blocks Affected by Reset and Powerdown Signals (Part 1 of 2)

Transceiver Block	pll_powerdown	rx_digitalreset	rx_analogreset	tx_digitalreset	tx_analogreset
PLLs					
CMU PLLs	Yes	—	—	—	—
ATX PLLs	Yes	—	—	—	—
Receiver Standard PCS					
Receiver Word Aligner	—	Yes	—	—	—
Receiver Deskew FIFO	—	Yes	—	—	—
Receiver Rate Match FIFO	—	Yes	—	—	—
Receiver 8B/10B Decoder	—	Yes	—	—	—
Receiver Byte Deserializer	—	Yes	—	—	—
Receiver Byte Ordering	—	Yes	—	—	—
Receiver Phase Compensation FIFO	—	Yes	—	—	—
Receiver 10G PCS					
Receiver Gear Box	—	Yes	—	—	—
Receiver Block Synchronizer	—	Yes	—	—	—
Receiver Disparity Checker	—	Yes	—	—	—
Receiver Descrambler	—	Yes	—	—	—
Receiver Frame Sync	—	Yes	—	—	—
Receiver 64B/66B Decoder	—	Yes	—	—	—
Receiver CRC32 Checker	—	Yes	—	—	—
Receiver FIFO	—	Yes	—	—	—
Receiver PMA					
Receiver Buffer	—	—	Yes	—	—
Receiver CDR	—	—	Yes	—	—
Receiver Deserializer	—	—	Yes	—	—
Transmitter Standard PCS					
Transmitter Phase Compensation FIFO	—	—	—	Yes	—
Byte Serializer	—	—	—	Yes	—
8B/10B Encoder	—	—	—	Yes	—
Transmitter Bit-Slip	—	—	—	Yes	—
Transmitter 10G PCS					
Transmitter FIFO	—	—	—	Yes	—
Transmitter Frame Generator	—	—	—	Yes	—
Transmitter CRC32 Generator	—	—	—	Yes	—

Table 3–1. Transceiver Blocks Affected by Reset and Powerdown Signals (Part 2 of 2)

Transceiver Block	pll_powerdown	rx_digitalreset	rx_analogreset	tx_digitalreset	tx_analogreset
Transmitter 64B/66B Encoder	—	—	—	Yes	—
Transmitter Scrambler	—	—	—	Yes	—
Transmitter Disparity Generator	—	—	—	Yes	—
Transmitter Gear Box	—	—	—	Yes	—
Transmitter PMA					
Transmitter CGB	—	—	—	—	Yes
Serializer	—	—	—	—	Yes
Transmitter Buffer	—	—	—	—	Yes

PHY IP Embedded Reset Controller

Figure 3–1 shows a block diagram of the transceiver PHY instance and the Transceiver Reconfiguration Controller interacting with the embedded reset controller.

Figure 3–1. Interaction Between the Transceiver PHY Instance, Transceiver Reconfiguration Controller, and the Embedded Reset Controller

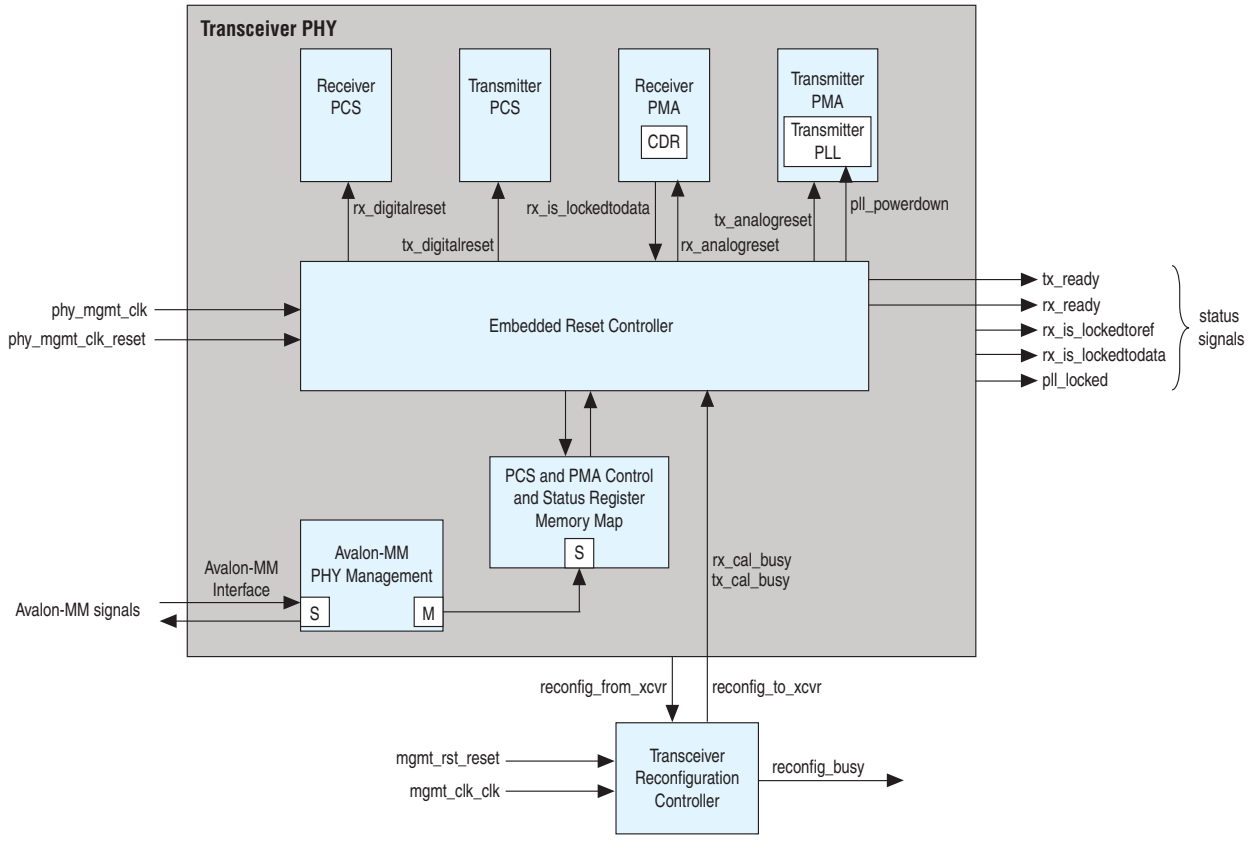


Table 3–2 lists the signals used by the transceiver PHY instance, Transceiver Reconfiguration Controller, and embedded reset controller.

Table 3–2. Signals Used by Transceiver PHY instance, Transceiver Reconfiguration Controller, and Embedded Reset Controller

Signal Name	Signal Type	Description
phy_mgmt_clk	Clock	Clock for the transceiver PHY IP
mgmt_clk_clk	Clock	Clock for the Transceiver Reconfiguration Controller
phy_mgmt_clk_reset	Reset	Reset for the transceiver PHY IP
mgmt_rst_reset	Reset	Reset for the Transceiver Reconfiguration Controller
pll_powerdown	Internal (1)	When the signal is high, the transmitter (TX) phase-locked loop (PLL) is held in reset
tx_analogreset	Internal (1)	When the signal is high, the TX PMA is held in reset
tx_digitalreset	Internal (1)	When the signal is high, the TX PCS is held in reset
rx_analogreset	Internal (1)	When the signal is high, the receiver (RX) PMA is held in reset
rx_digitalreset	Internal (1)	When the signal is high, the RX PCS is held in reset
tx_cal_busy (2)	Internal (1)	When the signal is high, TX calibration is active
rx_cal_busy (2)	Internal (1)	When the signal is high, RX calibration is active
reconfig_busy (2)	Status	When the signal is high, reconfiguration is active
pll_locked (3)	Status	When the signal is high, the TX PLL is locked
rx_is_lockedtoref (3)	Status	When the signal is high, the RX clock data recovery (CDR) is in the lock-to-reference (LTR) mode
rx_is_lockedtodata	Status	When the signal is high, the RX CDR is in the lock-to-data (LTD) mode
tx_ready	Status	When the signal is low, the TX datapath is in reset. A high indicates that the TX datapath is out of reset and ready for data transmission. The signal stays high until any of the reset signals are asserted.
rx_ready	Status	When the signal is low, the RX datapath is in reset. A high indicates that the RX datapath is out of reset and ready for data reception. The signal stays high until any of the reset signals are asserted.

Note to Table 3–2:

(1) Internal signals are those signals that are internal to the PHY IP embedded reset controller.
(2) These signals are synchronous to the reconfiguration controller input clock.
(3) These signals are asynchronous.

The embedded reset controller is level sensitive (active high).

The embedded reset controller automatically performs the entire transceiver reset sequence whenever the phy_mgmt_clk_reset signal is asserted. In case of loss-of-link or loss-of-data, the embedded reset controller asserts the appropriate reset signals.

You must monitor the status signals tx_ready and rx_ready. A high on these status signals indicates the transceiver is out of reset and ready for data transmission and reception.

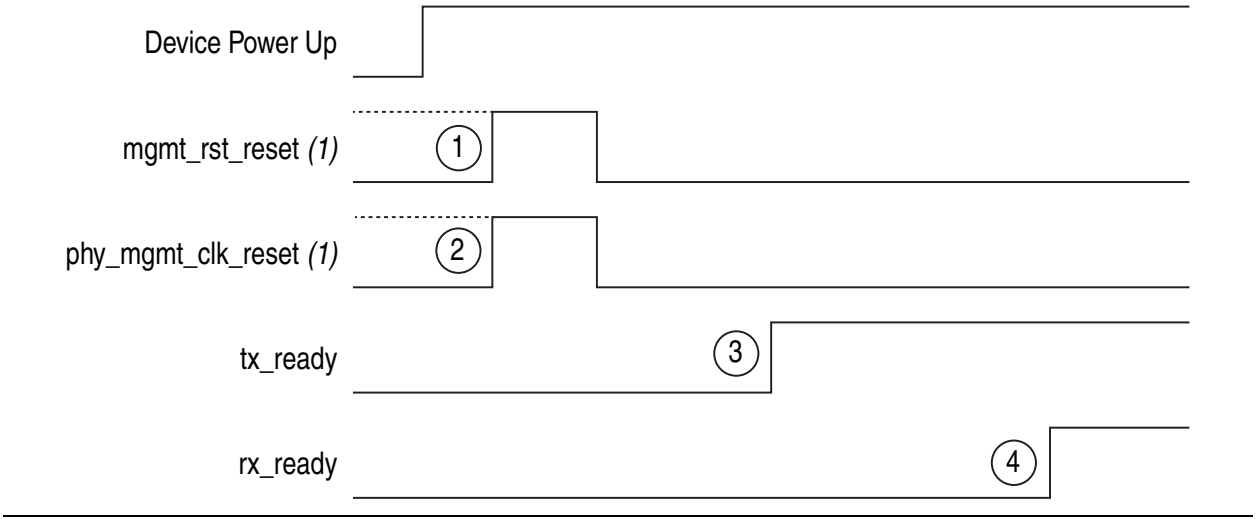
The mgmt_rst_reset signal needs to be asserted before or at the same time as phy_mgmt_clk_reset to start calibration.

Timing Diagram

Figure 3–2 shows the timing diagram of the embedded reset controller. Altera recommends re-initiating the entire reset sequence after channel or PLL reconfiguration. The following numbered list guides you through the sequence:

1. After device power-up, `mgmt_rst_reset` needs to be pulsed to start the calibration process. Alternatively, you can start off with `mgmt_rst_reset` asserted and de-assert it after device power-up.
2. `Phy_mgmt_clk_reset` needs to be pulsed at the same time or after `mgmt_rst_reset` to start the transceiver reset sequence. Alternatively, you can start off with `phy_mgmt_clk_reset` asserted and de-assert it after device power-up.
3. After all the transmitter calibration and reset sequence is complete the `tx_ready` status gets asserted, indicating that the transmitter is ready to transmit data.
4. After all the receiver calibration and reset sequence is complete the `rx_ready` status gets asserted, indicating that the receiver is ready to receive data.

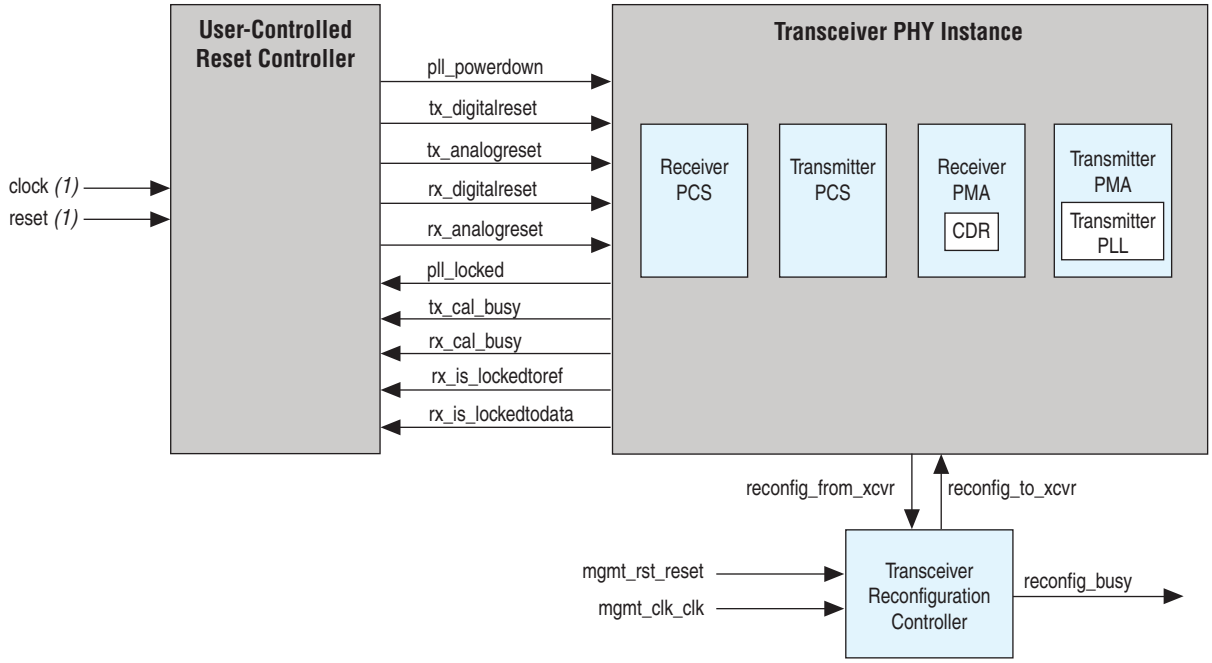
Figure 3–2. Timing Diagram for the Embedded Reset Controller



User-Controlled Reset Controller

Figure 3–3 shows a block diagram of the transceiver PHY instance and the Transceiver Reconfiguration Controller interacting with the User-Controlled Reset Controller.

Figure 3–3. Interaction Between the Transceiver PHY Instance, Transceiver Reconfiguration Controller, and the User-Controlled Reset Controller



Note to Figure 3–3:

(1) You can use `phy_mgmt_clk` and `phy_mgmt_clk_reset` as the clock and reset to the user-controlled reset logic.

Table 3–3 lists the signals used by the Transceiver PHY Instance, Transceiver Reconfiguration Controller, and User-Controlled Reset Controller.

Table 3–3. Signals Used by Transceiver PHY instance, Transceiver Reconfiguration Controller, and User-Controlled Reset Controller (Part 1 of 2)

Signal Name	Signal Type	Description
`mgmt_clk_clk`	Clock	Clock for the Transceiver Reconfiguration Controller
`mgmt_rst_reset`	Reset	Reset for the Transceiver Reconfiguration Controller
`pll_powerdown`	Control	Resets the TX PLL when asserted high
`tx_analogreset`	Control	Resets the TX PMA when asserted high
`tx_digitalreset`	Control	Resets the TX PCS when asserted high
`rx_analogreset`	Control	Resets the RX PMA when asserted high
`rx_digitalreset`	Control	Resets the RX PCS when asserted high
`reconfig_busy` (1)	Status	A high on this signal indicates that reconfiguration is active
`tx_cal_busy` (1)	Status	A high on this signal indicates that TX calibration is active
`rx_cal_busy` (1)	Status	A high on this signal indicates that RX calibration is active
`pll_locked` (2)	Status	A high on this signal indicates that the TX PLL is locked
`rx_is_lockedtoref` (2)	Status	A high on this signal indicates that the RX CDR is in the LTR mode

Table 3–3. Signals Used by Transceiver PHY instance, Transceiver Reconfiguration Controller, and User-Controlled Reset Controller (Part 2 of 2)

Signal Name	Signal Type	Description
rx_is_lockedtodata	Status	A high on this signal indicates that the RX CDR is in the LTD mode

Notes to Table 3–3:

(1) These signals are synchronous to the reconfiguration controller input clock.

(2) These signals are asynchronous.

If you disable the embedded reset controller, you must implement your own reset controller logic. You must consider the following:

- The user-controlled reset controller can be level sensitive (active high)
- The user-controlled reset controller does not depend on phy_mgmt_clk_reset
- You are responsible for providing a clock and reset to the reset controller logic
- The internal signals of the PHY IP embedded reset controller are exposed as ports

Altera also offers a Transceiver PHY reset controller that you can use instead of writing the complete user-controlled reset logic. The Transceiver PHY reset controller is only available for non-protocol Transceiver PHY IP cores (Custom, Low Latency, Deterministic Latency, and Native PHY IP). Refer to the "Transceiver PHY Reset Controller IP Core" section of the *Altera Transceiver PHY IP Core User Guide*.

Timing Diagram During Device Power-Up and After Reconfiguration

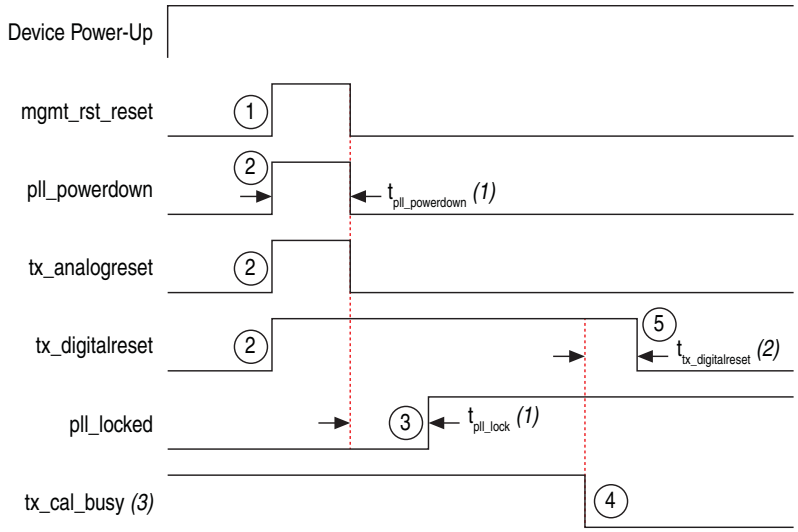
Figure 3–4 shows the reset sequence of the user-controlled reset controller during device power-up and after reconfiguration for the transmitter. Altera recommends re-initiating the entire reset sequence after channel or PLL reconfiguration. The following numbered list guides you through the entire reset sequence:

1. After device power-up mgmt_rst_reset needs to be pulsed to start the calibration process. Alternatively, you can start off with mgmt_rst_reset asserted and de-assert it after device power-up.
2. pll_powerdown, tx_analogreset, and tx_digitalreset need to be pulsed at the same time or after mgmt_rst_reset to start the transmitter reset sequence.
3. After the transmitter PLL locks, the pll_locked status gets asserted after t_{pll_lock}.
4. After the transmitter calibration completes, the tx_cal_busy status gets de-asserted. This could happen before or after the pll_locked gets asserted, depending on the different transmitter calibrations enabled.

During any ATX PLL calibration make sure the pll_powerdown is not asserted.

5. tx_digitalreset can be de-asserted $t_{tx\_digitalreset}$ after all the three gating conditions are removed, which include:
   - pll_powerdown is de-asserted
   - pll_locked is asserted
   - tx_cal_busy is de-asserted

Figure 3–4. Timing Diagram for the User-Controlled Reset Controller During Device Power-Up and After Reconfiguration for the Controller



Notes to Figure 3–4:

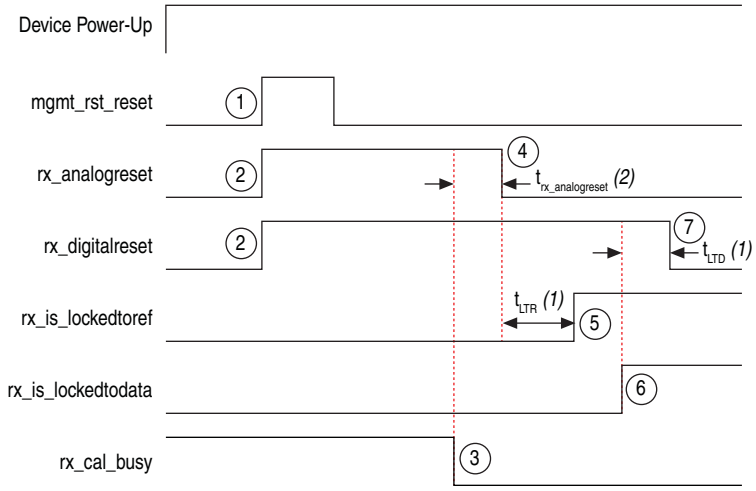
(1) t_{pll_lock} and $t_{pll\_powerdown}$, are pending characterization. For more information, refer to the *Stratix V Device Datasheet*.
(2) $t_{tx\_digitalreset}$ = 20ns
(3) tx_cal_busy signal is high only during power-up calibration and is low for any other dynamic reconfiguration.

Figure 3–5 shows the reset sequence of the user-controlled reset controller during device power-up and after reconfiguration for the receiver. Altera recommends re-initiating the entire reset sequence after channel or PLL reconfiguration. The following numbered list guides you through the entire reset sequence:

1. After device power-up mgmt_rst_reset needs to be pulsed to start the calibration process. Alternatively you can start off with mgmt_rst_reset asserted and de-assert it after device power-up.
2. rx_analogreset and rx_digitalreset need to be pulsed at the same time or after mgmt_rst_reset to start the receiver reset sequence.
3. After the receiver calibration completes, the rx_cal_busy status gets de-asserted.
4. tx_analogreset can be de-asserted $t_{rx\_analogreset}$ after rx_cal_busy gets de-asserted
5. rx_is_lockedtoref status gets asserted after t_{LTR}.
6. rx_is_lockedtodata status gets asserted once the CDR switches from lock-to-reference to lock-to-data mode.

7. `rx_digitalreset` can be de-asserted t_{LTD} after `rx_is_lockedtodata` gets asserted. Also make sure the following two gating conditions are removed:
   - `rx_analogreset` is de-asserted
   - `rx_cal_busy` is de-asserted

Figure 3–5. Reset Sequence of the User-Controlled Reset Controller During Device Power-Up and After Reconfiguration for the Receiver



Notes to Figure 3–5:

(1) t_{LTR} and t_{LTD} are pending characterization. For more information, refer to the *Stratix V Device Datasheet*.

(2) $t_{rx\_analogreset}$ = 40ns.

CDR Modes During the Reset Sequence

Whether you are using the embedded reset controller or the user-controlled reset controller, the CDR can be in either of the following modes:

- Automatic Lock Mode - Default mode
- Manual Lock Mode

Reset Sequence for CDR in Manual Lock Mode

The two control signals to enable the CDR in manual lock mode are `rx_set_locktoref` and `rx_set_locktodata`.

Table 3–4 lists the control settings for the CDR in manual lock mode.

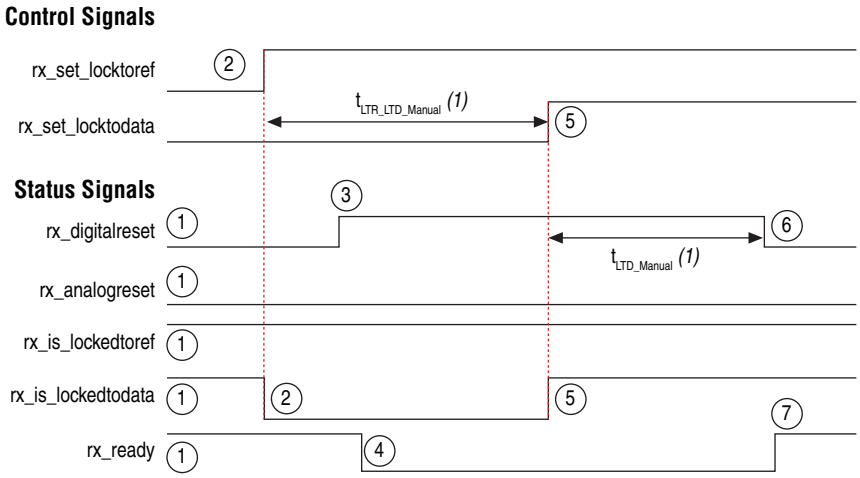
Table 3–4. Control Settings for the CDR in Manual Lock Mode

rx_set_locktoref	rx_set_locktodata	CDR Lock Mode
0	0	Automatic
1	0	Manual-RX CDR LTR
X	1	Manual-RX CDR LTD

Figure 3–6 shows the timing diagram of the transceiver reset sequence for the CDR in manual lock mode. The following numbered list guides you through the steps to put the CDR in the manual lock mode.

1. Make sure that the calibration is complete and the transceiver goes through the initial reset sequence. The `rx_digitalreset` and `rx_analogreset` signals should be low. The `rx_is_lockedtoref` is a don't care and can be either high or low. The rx_is_lockedtodata and rx_ready signals should be high, indicating that the transceiver is out of reset. Alternatively, you can start directly with the CDR in manual lock mode after the calibration is complete.
2. Assert the `rx_set_locktoref` signal high to switch the CDR to the lock-to-reference mode. The `rx_is_lockedtodata` status signal gets de-asserted.
3. Assert the `rx_digitalreset` signal high if you are using the user-controlled reset. In case of embedded reset controller the `rx_digitalreset` is a status signal that automatically gets asserted.
4. After the `rx_digitalreset` signal gets asserted, the `rx_ready` status signal gets de-asserted.
5. Assert the `rx_set_locktodata` signal high after $t_{LTR\_LTD\_manual}$ to switch the CDR to the lock-to-data mode. The `rx_is_lockedtodata` status signal gets asserted. The `rx_is_lockedtoref` status signal can be a high or low and can be ignored.
6. De-assert the `rx_digitalreset` signal after t_{LTD_Manual}.
7. After the `rx_digitalreset` signal is de-asserted, the `rx_ready` status signal gets asserted if you are using the embedded reset controller, indicating that the receiver is now ready to receive data with the CDR in manual mode.

Figure 3–6. Timing Diagram for the Transceiver Reset Sequence for CDR in Manual Lock Mode



Note to Figure 3–6:

(1) t_{LTD_Manual} and $t_{LTR\_LTD\_Manual}$ are pending characterization. For more information, refer to the *Stratix V Device Datasheet*.

Document Revision History

Table 3–5 lists the revision history for this chapter.

Table 3–5. Document Revision History

Date	Version	Changes
June 2012	3.3	■ Updated for the Quartus II software version 12.0. ■ Revised Figure 3–2, Figure 3–4, and Figure 3–6. ■ Added new Table 3–1. ■ Added new steps describing wave forms. ■ Deleted power-up Figures 3-2 and 3-5 and changed text accordingly.
February 2012	3.2	■ Minor changes to Figure 3-4. ■ Minor text edits.
December 2011	3.1	■ Changed "Stratix V Device Power-Up" to "Device Power-Up" in Figure 3–2, Figure 3–3, Figure 3–5, Figure 3–6, and Figure 3–7. ■ Minor text edits.
November 2011	3.0	■ Added "Embedded Reset Controller" and "User-Controlled Reset Controller" sections. ■ Updated Figure 3–2, Figure 3–3, Figure 3–5, Figure 3–6, and Figure 3–7. ■ Added the `tx_analogreset` signal. ■ Minor text edits.
May 2011	2.0	■ Added sections for CDR reset sequence in automatic and manual lock mode. ■ Removed sections for PCIe and non-PCIe reset sequences. ■ Chapter moved to Volume 3.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

4. Transceiver Configurations in Stratix V Devices

SV52005-2.3

This chapter provides the transceiver channel datapath, clocking guidelines, channel placement guidelines, and a brief description of protocol features supported in each transceiver configuration for Stratix® V devices.

Stratix V devices have dedicated transceiver physical coding sublayer (PCS) and physical medium attachment (PMA) circuitry and a dedicated PHY IP to support the following communication protocols:

- "10GBASE-R" on page 4–2
- "Interlaken" on page 4–8
- "PCI Express (PCIe)—Gen1, Gen2, and Gen3" on page 4–15
- "CPRI and OBSAI—Deterministic Latency Protocols" on page 4–40
- "XAUI" on page 4–46

Support for other communication protocols or user-defined protocols can be enabled with the following PHY IPs:

- Custom PHY IP using the standard PCS in a custom datapath
- Low Latency PHY IP using the standard or 10G PCS in a low latency datapath configuration
- Native PHY IP using PMA direct configuration

The transceiver configuration datapaths for the standard PCS and 10G PCS blocks in Stratix V GX devices are described in the following sections:

- "Standard PCS Configurations" on page 4–54
- "10G PCS Configurations" on page 4–69

For information about the Native PHY IP, refer to "Stratix V GX Native PHY IP" on page 4–78.

For information about Stratix V GT device configurations, refer to "Stratix V GT Device Configurations" on page 4–79.

For a complete list of serial protocols supported by Stratix V devices, refer to the *Upcoming Stratix V Device Features* document.

Use this chapter along with the *Altera Transceiver PHY IP Core User Guide* to implement your intended protocol links in Stratix V devices.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Table 4–1 lists the Altera® PHY IP Core instance names that you must instantiate for each supported transceiver configuration.

Table 4–1. Altera PHY IP Core Names

Transceiver Configuration	Altera PHY IP Core
10GBASE-R	10GBASE-R PHY
Interlaken	Interlaken PHY
PCI Express® (PCIe®)	PCI Express PHY (PIPE)
XAUI	XAUI PHY
CPRI/OBSAI	Deterministic PHY
All others	Custom, Low Latency, or Native PHY
User-defined protocol	Custom, Low Latency, or Native PHY

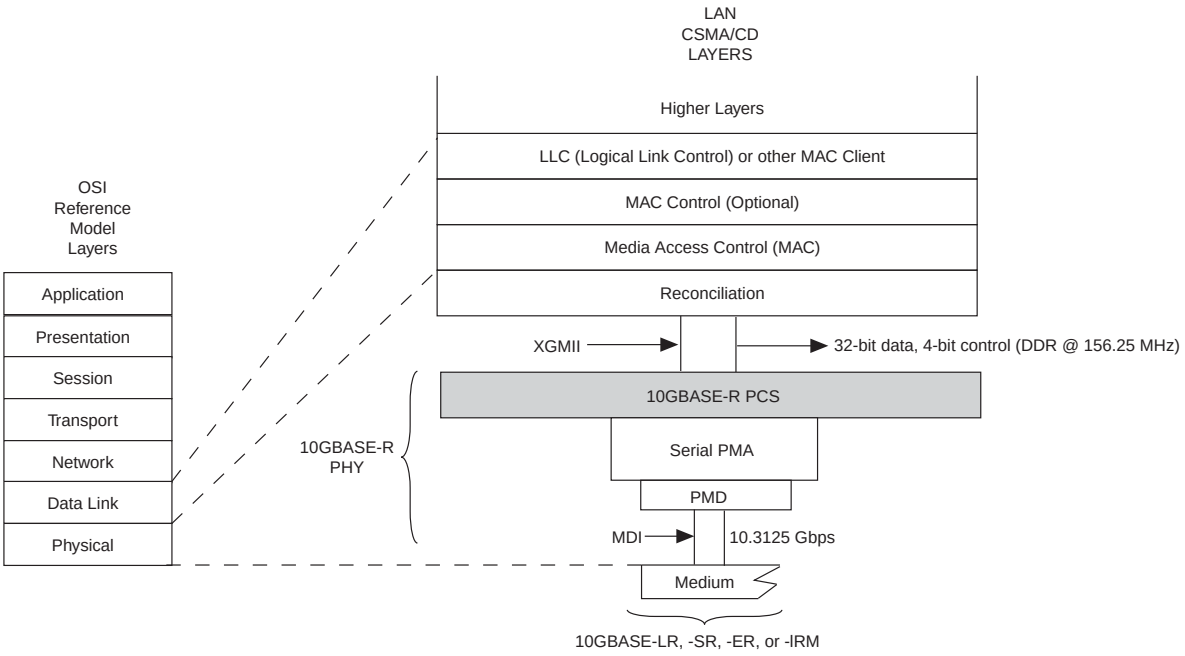
10GBASE-R

This section describes 10GBASE-R link implementation using Stratix V transceivers. The section provides the transceiver channel datapath and clocking when configured in a 10GBASE-R configuration.

10GBASE-R is a specific physical layer implementation of the 10 Gigabit Ethernet link defined in clause 49 of the IEEE 802.3-2008 specification. As shown in Figure 4–1, the 10GBASE-R PHY uses the XGMII interface to connect to the IEEE802.3 media access control (MAC) and reconciliation sublayer (RS). The IEEE 802.3-2008 specification requires each 10GBASE-R link to support a 10 Gbps data rate at the XGMII interface and a 10.3125 Gbps serial line rate with 64B/66B encoding.

Figure 4–1 shows the relationship between the 10GBASE-R PHY and other sublayers in the OSI reference model.

Figure 4–1. 10GBASE-R PHY Connection to IEEE802.3 MAC and RS



In the MegaWizard™ Plug-In Manager, you can implement a 10GBASE-R link by instantiating the **10GBASE-R PHY IP** core under **Ethernet** in the Interfaces menu.

For more information about the 10GBASE-R PHY IP, refer to the *10GBASE-R PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Transceiver Datapath Configuration

Figure 4–2 shows the transceiver blocks and settings enabled in a 10GBASE-R configuration. The blocks shown as "Disabled" are not used, but incur latency. The blocks shown as "Bypassed" are not used and do not incur latency.

Figure 4–2. 10GBASE-R Datapath Configuration

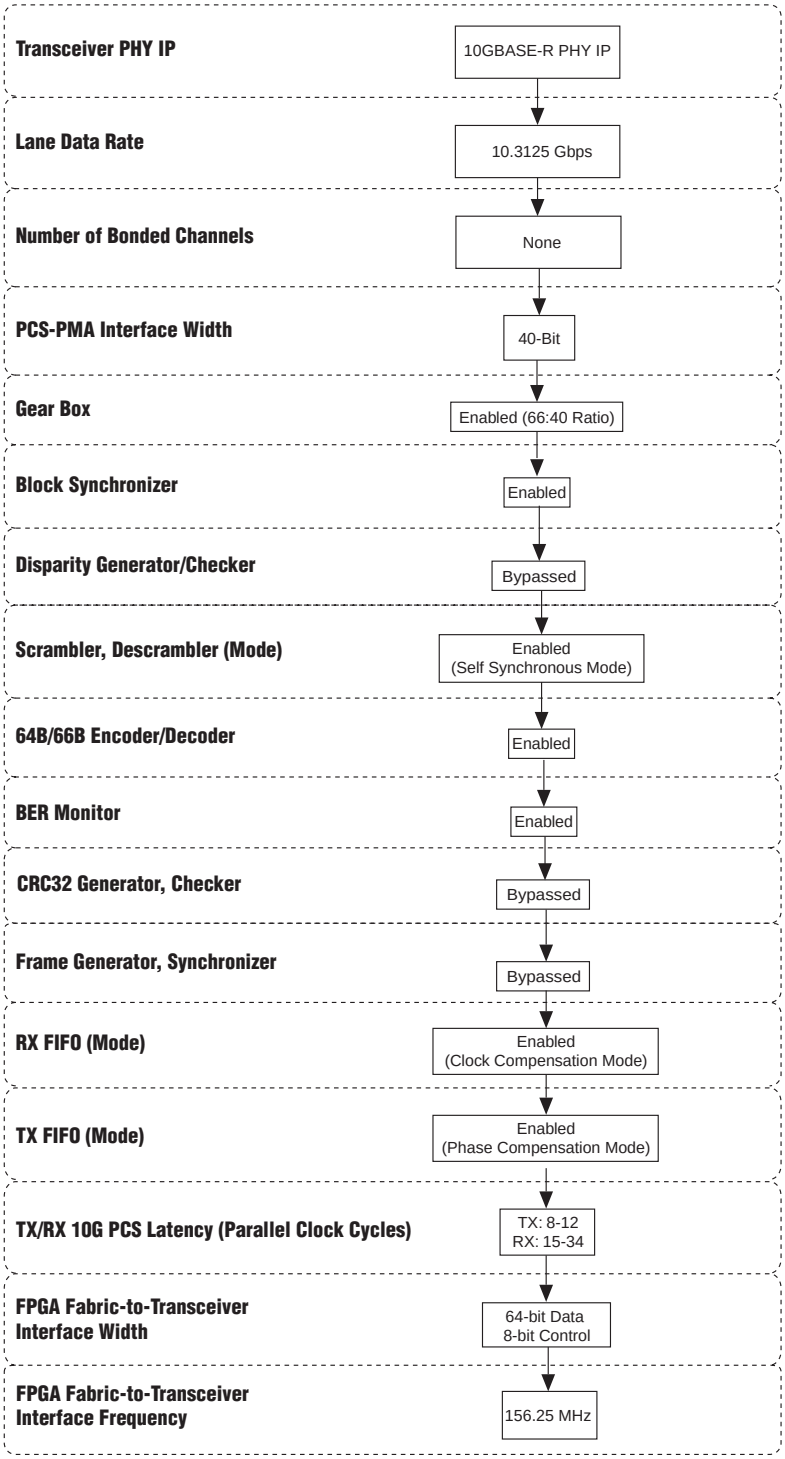
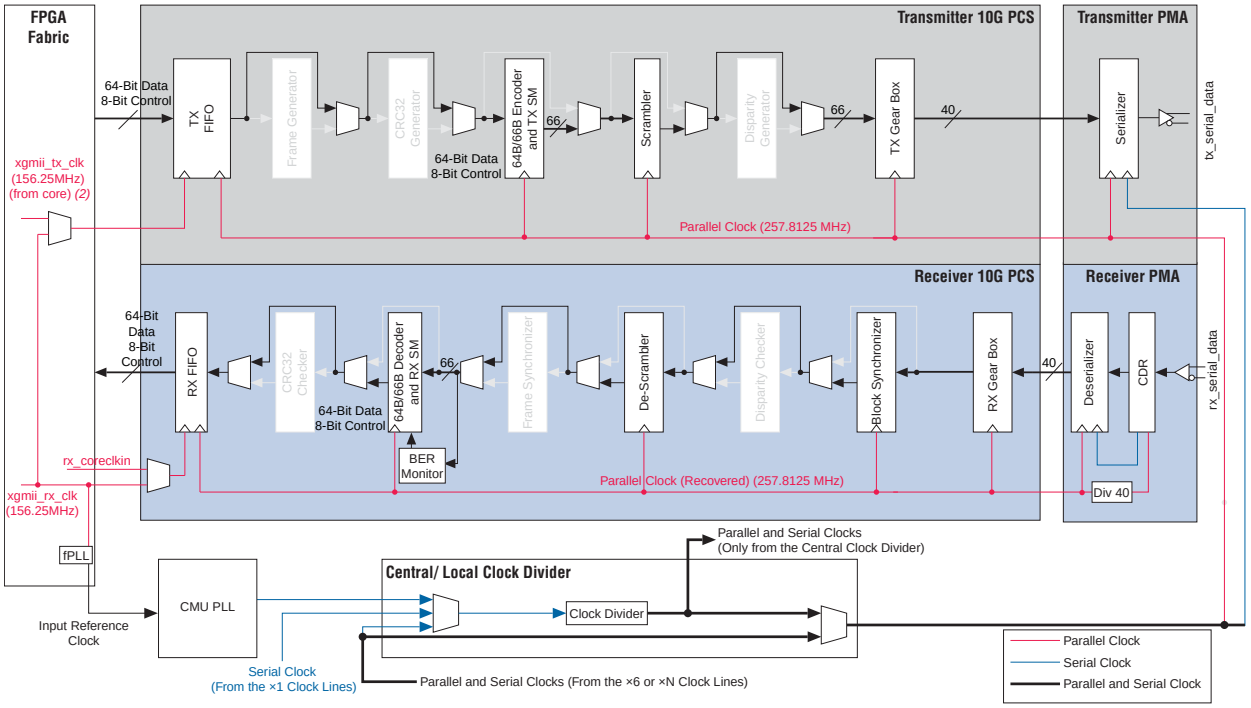


Figure 4–3 shows the transceiver datapath in a 10GBASE-R configuration.

Figure 4–3. Transceiver Channel Datapath for a 10GBASE-R Configuration *(1)*



Notes to Figure 4–3:

(1) The x1 clock line can be driven by either the ATX or CMU PLL.

(2) The `xgmii_tx_clk` requires 0 ppm difference from the `xgmii_rx_clk` or the Input Reference Clock.

Supported Features

The following sections describe the features supported by Stratix V transceivers in a 10GBASE-R configuration.

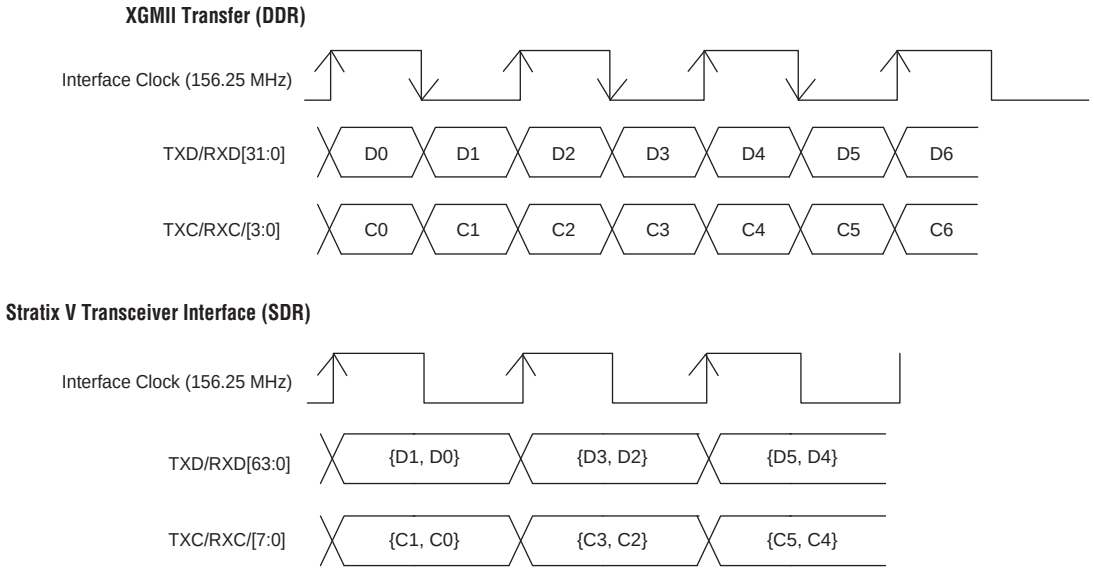
For details about 10GBASE-R PHY IP control and status signals associated with each feature, refer to the *10GBASE-R PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

64-Bit Single Data Rate (SDR) Interface to the MAC/RS

Clause 46 of the IEEE 802.3-2008 specification defines the XGMII interface between the 10GBASE-R PCS and the Ethernet MAC/RS. The XGMII interface defines the 32-bit data and 4-bit wide control character clocked between the MAC/RS and the PCS at both the positive and negative edge (double data rate – DDR) of the 156.25 MHz interface clock.

Stratix V transceivers do not support the XGMII interface to the MAC/RS as defined in the IEEE 802.3-2008 specification. Instead, they support a 64-bit data and 8-bit control SDR interface between the MAC/RS and the PCS, as shown in Figure 4–4.

Figure 4–4. XGMII Interface (DDR) versus Stratix V Transceiver Interface (SDR) for 10GBASE-R



64B/66B Encoding/Decoding

Stratix V transceivers in a 10GBASE-R configuration support 64B/66B encoding and decoding as specified in Clause 49 of the IEEE802.3-2008 specification. The 64B/66B encoder receives 64-bit data and 8-bit control code from the transmitter FIFO and converts it into 66-bit encoded data. The 66-bit encoded data contains two overhead sync header bits that the receiver PCS uses for block synchronization and bit-error rate (BER) monitoring.

The 64B/66B encoding also ensures enough transitions on the serial data stream for the receiver clock data recovery (CDR) to maintain its lock on the incoming data.

Transmitter and Receiver State Machines

Stratix V transceivers in a 10GBASE-R configuration implement the transmitter and receiver state diagrams shown in Figure 49-14 and Figure 49-15 of the IEEE802.3-2008 specification.

Besides encoding the raw data specified in the 10GBASE-R PCS, the transmitter state diagram performs functions such as transmitting local faults (LBLOCK_T) under reset, as well as transmitting error codes (EBLOCK_T) when the 10GBASE-R PCS rules are violated.

Besides decoding the incoming data specified in the 10GBASE-R PCS, the receiver state diagram performs functions such as sending local faults (LBLOCK_R) to the MAC/RS under reset and substituting error codes (EBLOCK_R) when the 10GBASE-R PCS rules are violated.

Block Synchronizer

The block synchronizer in the receiver PCS determines when the receiver has obtained lock to the received data stream. It implements the lock state diagram shown in Figure 49-12 of the IEEE 802.3-2008 specification.

The block synchronizer provides a status signal to indicate whether it has achieved block synchronization or not.

Self-Synchronous Scrambling/Descrambling

The scrambler/descrambler blocks in the transmitter/receiver PCS implements the self-synchronizing scrambler/descrambler polynomial 1 + x39 + x58, as described in clause 49 of the IEEE 802.3-2008 specification. The scrambler/descrambler blocks are self-synchronizing and do not require an initialization seed. Barring the two sync header bits in each 66-bit data block, the entire payload is scrambled or descrambled.

BER Monitor

The BER monitor block in the receiver PCS implements the BER monitor state diagram shown in Figure 49-13 of the IEEE 802.3-2008 specification. The BER monitor provides a status signal to the MAC whenever the link BER threshold is violated.

The 10GBASE-R PHY IP core provides a status flag to indicate a high BER whenever 16 synchronization header errors are received within a 125 µs window.

Clock Compensation

The receiver FIFO in the receiver PCS datapath compensates up to ±100 ppm difference between the remote transmitter and the local receiver. The receiver FIFO does so by inserting Idles (/I/) and deleting Idles (/I/) or Ordered Sets (/O/), depending on the ppm difference.

Idle Insertion

The receiver FIFO inserts eight /I/ codes following an /I/ or /O/ to compensate for clock rate disparity.

Idle (/I/) or Sequence Ordered Set (/O/) Deletion

The receiver FIFO deletes either four /I/ codes or ordered sets (/O/) to compensate for the clock rate disparity. The receiver FIFO implements the following IEEE802.3-2008 deletion rules:

- Deletes the lower four /I/ codes of the current word when the upper four bytes of the current word do not contain a Terminate /T/ control character.
- Deletes the upper four /I/ codes of the current word when the previous word's lower four bytes do not contain a Terminate /T/ control character.
- Deletes one /O/ ordered set only when the receiver FIFO receives two consecutive /O/ ordered sets.

Transceiver Clocking

This section describes the transceiver clocking for the 10GBASE-R protocol supported in Stratix V devices.

Figure 4–3 on page 4–5 shows transceiver clocking in a 10GBASE-R configuration.

The CMU PLL or the auxiliary transmit (ATX) PLLs in a transceiver bank generate the transmitter serial and the fractional PLL for the parallel clocks for the 10GBASE-R channels. Table 4–2 lists the configuration details.

Table 4–2. Input Reference Clock Frequency and Interface Speed Specifications for 10GBASE-R Configurations

Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Frequency (MHz)
644.53125, 322.265625	64-bit data, 8-bit control	156.25

Interlaken

This section describes Interlaken link implementation with Stratix V transceivers. The section provides the transceiver channel datapath, clocking, and channel placement guidelines when configured for Interlaken.

Interlaken is a scalable, chip-to-chip interconnect protocol that enables transmission speeds from 10 to more than 100 Gbps. Stratix V devices support a transmission speed of up to 10.3125 Gbps per lane in an Interlaken configuration. All the PCS blocks in the Interlaken configuration conform to the Interlaken Protocol Definition, Rev 1.2.

In the MegaWizard Plug-In Manager, you can implement an Interlaken link by instantiating the **Interlaken PHY IP** under **Interlaken** in the Interfaces menu.

For more information about the Interlaken PHY IP, refer to the *Interlaken PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Transceiver Datapath Configuration

Figure 4–5 shows the transceiver blocks and settings enabled in an Interlaken configuration. Blocks shown as “Disabled” are not used but incur latency. Blocks shown as “Bypassed” are not used and do not incur any latency. The maximum data rates and frequencies shown in Figure 4–5 are for the fastest speed grade devices.

Figure 4–5. Interlaken Datapath Configuration

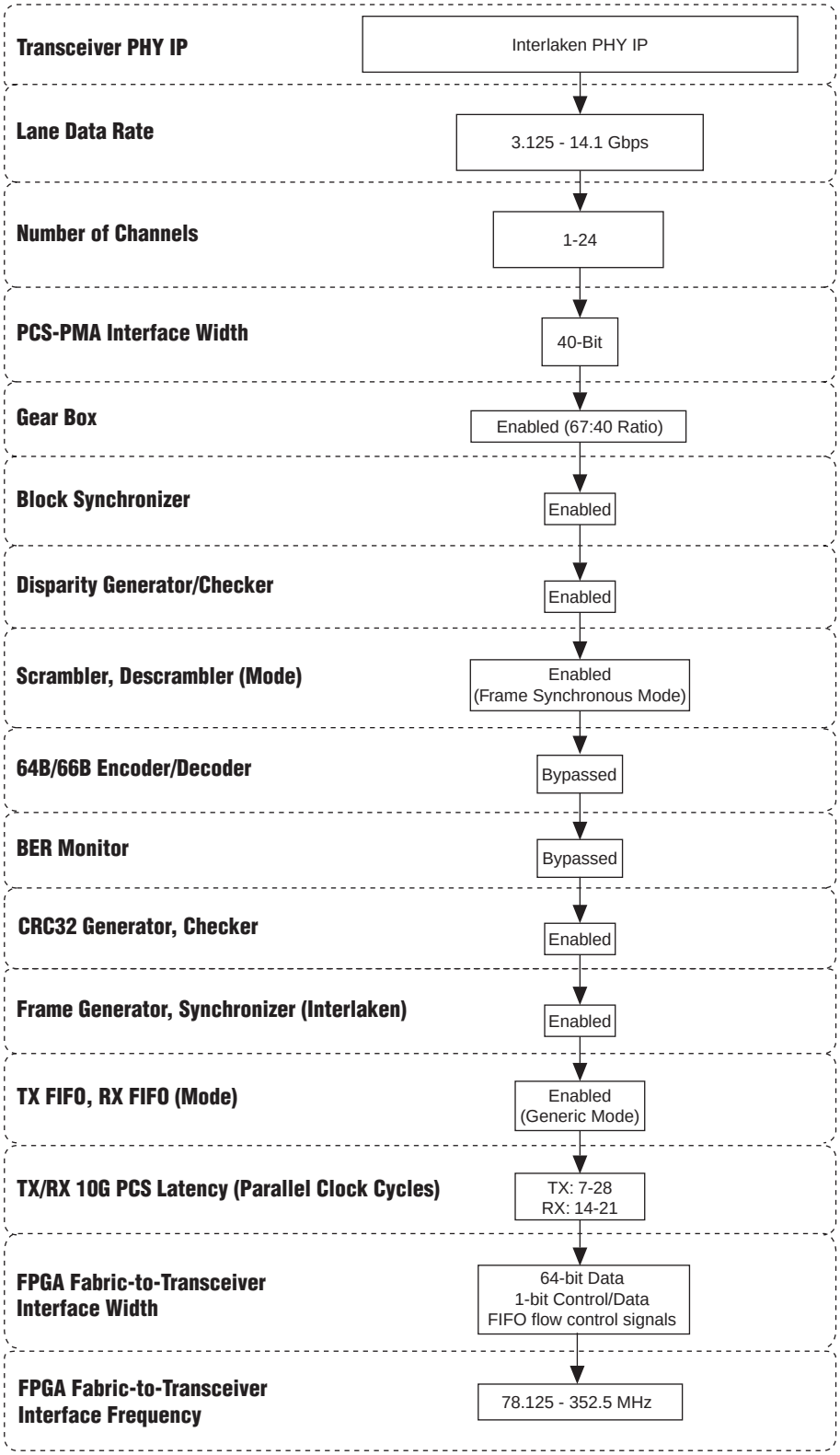
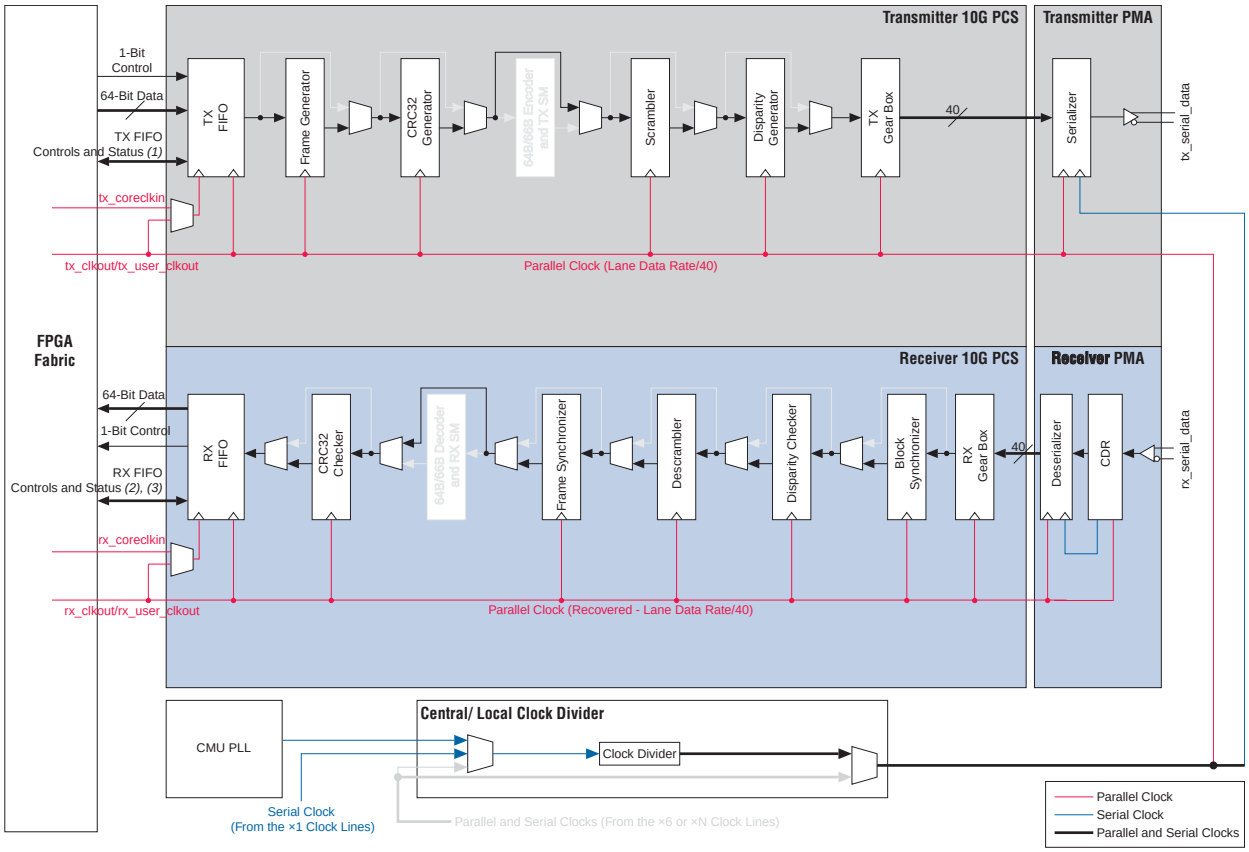


Figure 4–6 shows the PCS and PMA blocks used in the transceiver datapath for Interlaken configurations.

Figure 4–6. Transceiver Channel Datapath for Interlaken Configuration



Notes to Figure 4–6:

(1) TX FIFO Control and Status (transmit backpressure and datavalid, synchronization done)
(2) RX FIFO Control (receive FIFO read enable and datavalid
(3) RX FIFO Status (receive FIFO overflow and partially empty)

Supported Features

Table 4–3 lists the framing layer functions supported in Stratix V devices and defined in the Interlaken Protocol Definition, Rev 1.2.

Table 4–3. Supported Features in Interlaken Configuration (Part 1 of 2)

Feature	Supported
Metaframe generation and payload insertion	Yes
Block synchronization (word alignment) and metaframe synchronization (frame synchronization)	Yes
64B/67B framing	Yes
±96 bits disparity maintenance	Yes
Frame synchronous scrambling and descrambling	Yes
Diagnostic word generation	Yes

Table 4–3. Supported Features in Interlaken Configuration (Part 2 of 2)

Feature	Supported
Framing Layer Control Word Forwarding	Yes
CRC-32 generation and checking of lane data integrity	Yes
Multi-lane deskew alignment	No
Transmit and receive FIFO backpressure control and handshake	Yes

For more information about Interlaken PHY IP control and status signals associated with each feature, refer to the *Interlaken PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Block Synchronizer

The block synchronizer in the receiver PCS achieves and maintains a 64B/67B word boundary lock. This block searches for valid synchronization header bits within the data stream and achieves lock after 64 consecutive legal synchronization patterns are found. After a 64B/67B word boundary lock is achieved, the block synchronizer continuously monitors and flags for invalid synchronization header bits. If 16 or more invalid synchronization header bits are found within 64 consecutive word boundaries, the block synchronizer de-asserts the lock state and searches again for valid synchronization header bits.

The block synchronizer implements the flow diagram shown in Figure 13 of Interlaken Protocol Definition v1.2 and provides the word lock status to the FPGA fabric.

64B/67B Frame Generator

The transmit frame generator implements 64B/67B encoding, as explained in Interlaken Protocol Definition v1.2. The Interlaken metaframe generator synchronously generates the framing layer control words, frame synchronizer, scrambler state, skip words, and diagnostic word, and maps the transmitter data into the payload of the metaframes. The metaframe length is programmable from 5 to a maximum value of 8191, 8-byte words.

Ensure that the metaframe length is programmed to the same value for both the transmitter and receiver.

Frame Synchronizer

The receive frame synchronizer delineates the metaframe boundaries and searches for each of the framing layer control words: Synchronization, Scrambler State, Skip, and Diagnostic. When four consecutive synchronization words have been identified, the frame synchronizer achieves the frame locked state. Subsequent metaframes are then checked for valid synchronization and scrambler state words. If four consecutive invalid synchronization words or three consecutive mismatched scrambler state words are received, the frame synchronizer loses frame lock. In addition, the frame synchronizer provides a receiver metaframe lock status to the FPGA fabric.

Running Disparity

The disparity generator inverts the sense of bits in each transmitted word to maintain a running disparity of ± 96 bit boundary. It supplies a framing bit in bit position 66 as explained in Table 4 of Interlaken Protocol Definition Revision 1.2. The framing bit enables the disparity checker to identify whether bits[63:0] for that word are inverted.

Frame Synchronous Scrambling/Descrambling

The scrambler/descrambler block in the transmitter/receiver PCS implements the scrambler/descrambler polynomial $x^{58} + x^{39} + 1$ per Interlaken Protocol Definition Revision 1.2. Synchronization and Scrambler State Words, as well as the 64B/67B framing bits are not scrambled/descrambled. The Interlaken PHY IP core automatically programs random linear feedback shift register (LFSR) initialization seed values per lane.

The receiver PCS synchronizes the scrambler with the metaframe as described in the state flow shown in Figure 1 of Interlaken Protocol Definition Revision 1.2.

The frame synchronizer features a whole set of error and performance monitoring ports to the FPGA fabric interface and register status bits when using the Avalon® Memory-Mapped Management Interface. A receiver ready port, frame lock status, and cyclic redundancy check (CRC)-32 error detection port is available to the FPGA fabric. The Avalon Memory-Mapped Management Interface provides additional functionality with word boundary lock, frame lock status, synchronization word error detection, scrambler mismatch error, and CRC-32 error detection status register bits.

Skip Word Insertion

The frame generator generates the mandatory skip words with every metaframe following the scrambler state word for clock rate compensation and generates additional skip words based on the transmitter FIFO capacity state.

Skip Word Deletion

The frame synchronizer does not delete skip words. Instead, the frame synchronizer forwards the skip words it receives to the MAC layer so the MAC can maintain deskew alignment.

Diagnostic Word Generation and Checking of Lane Data Integrity (CRC-32)

The CRC-32 generator calculates the CRC for each metaframe and appends it to the diagnostic word of the metaframe. An optional CRC-32 error flag is also provided to the FPGA fabric.

Framing Layer Control Word Forwarding

The four metaframe framing layer control words—Synchronization, Scrambler State, Skip, and Diagnostic Words—are not deleted but forwarded to the MAC layer. This action enables the MAC layer to employ multi-lane deskew alignment within the FPGA fabric.

Multi-Lane Deskew Alignment

The Interlaken PHY IP does not support multi-lane deskew alignment. You must use either the multi-lane deskew alignment state machine or the Altera Interlaken MegaCore® function within the FPGA fabric.

For details about the Interlaken MegaCore function, refer to the *Interlaken MegaCore Function User Guide*.

Transmit and Receive FIFO Control and Status

The Interlaken PCS configures the transmit and receive FIFOs in elastic buffer mode. In this mode of operation, a lane synchronization, backpressure and FIFO control, and status port signals are provided to the MAC layer for handshaking.

Transceiver Multi-Lane Bonding and Transmit Skew

A soft-bonding IP is used for Interlaken bonding in Stratix V transceivers. The transceiver clocking in each lane is configured as non-bonded. For multi-lane designs, a dedicated PLL reference clock pin that is equidistant from the transmit PLLs in each bank must be selected. You must tightly match lane board traces to minimize lane-to-lane skew.

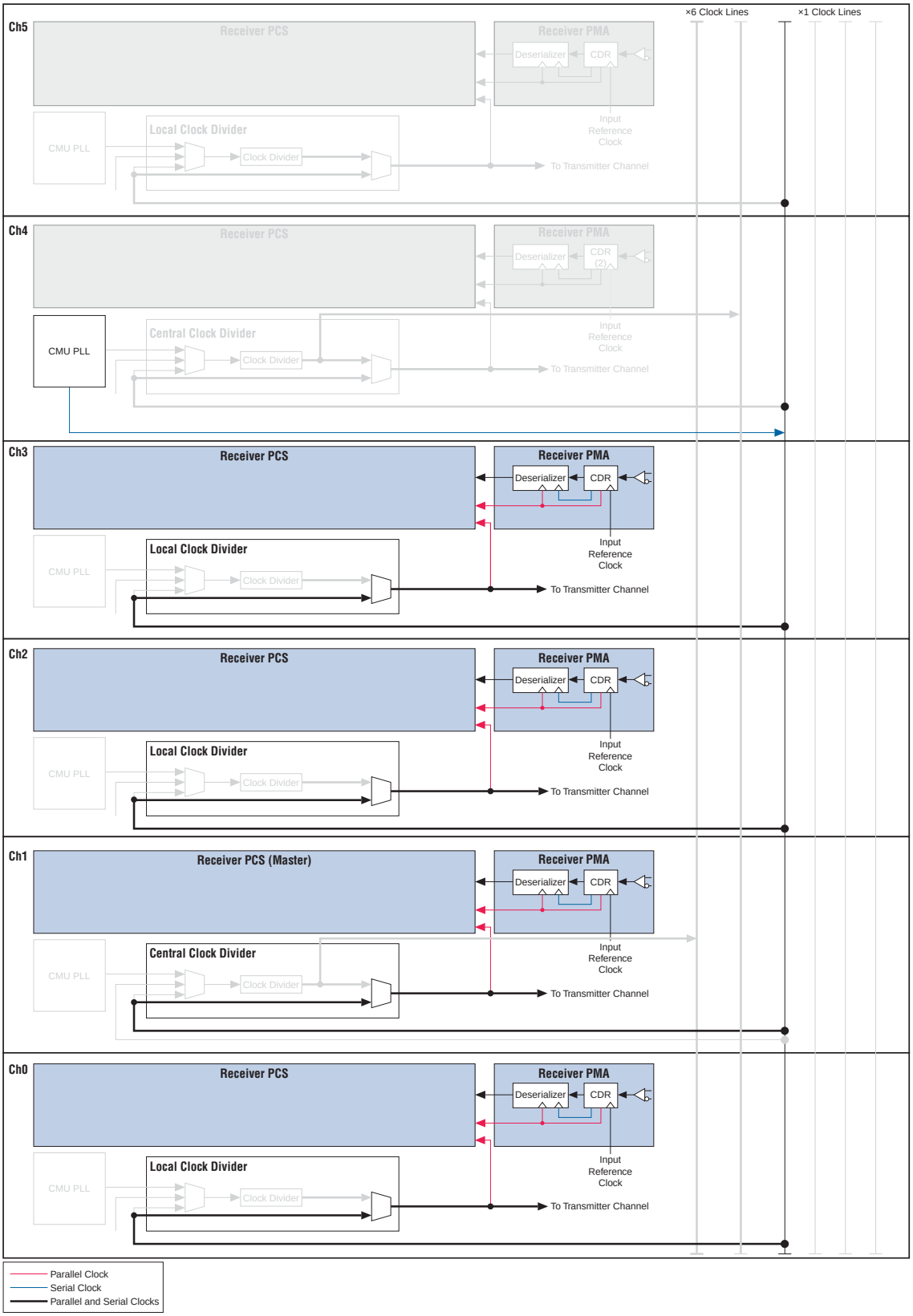
Transceiver Clocking

This section describes the transceiver clocking for the Interlaken protocol supported in Stratix V devices.

Transceiver Clocking

Figure 4–7 shows the clocking resources available in a four-lane Interlaken transceiver configuration.

Figure 4–7. Interlaken Four-Lane Configuration



A CMU PLL may provide a clock for up to five Interlaken lanes within a transceiver bank. If an ATX PLL is used, the PLL can clock up to six Interlaken lanes in a transceiver bank.

To enable the ATX PLL, you must select **ATX PLL** for the **PLL type** parameter in the **Interlaken PHY IP**.

PCI Express (PCIe)—Gen1, Gen2, and Gen3

The PCIe specification (version 3.0) provides implementation details for a PCIe-compliant physical layer device at Gen1 (2.5 Gbps), Gen2 (5 Gbps), and Gen3 (8 Gbps) signaling rates.

Stratix V devices have built-in PCIe hard IP blocks to implement the PHY MAC layer, data link layer, and transaction layer of the PCIe protocol stack. Up to four PCIe hard IP block reside within a Stratix V device. If you enable the PCIe hard IP block, the transceiver interfaces with the hard IP block. Otherwise, the transceiver interfaces directly with the FPGA fabric through the PIPE interface.

You can configure the Stratix V transceivers in a PCIe functional configuration using one of the following methods:

- Stratix V hard IP for PCI Express
- PHY IP core for PCI Express (PIPE)

Table 4–4 shows which Stratix V transceivers in a PCIe functional configuration are supported using the two methods.

Table 4–4. Support for Stratix V Transceivers

Support	Stratix V Hard IP for PCI Express	PHY IP Core for PCI Express (PIPE)
PCIe Hard IP Embedded HardCopy Block	Yes	—
Gen1 and Gen2 data rates	Yes	Yes
Gen3 data rate	Yes	—
Transceiver interface	Hard IP	PIPE 2.0

For details about PCIe hard IP, refer to the *Stratix V Hard IP for PCI Express User Guide*.

To enable PCIE Gen1 and Gen2 operations using PIPE mode, refer to the *PHY IP Core for PCI Express (PIPE)* chapter in the *Altera Transceiver PHY IP Core User Guide*.

In the MegaWizard Plug-In Manager, you can implement the PHY IP Core for PCI Express (PIPE) configuration by instantiating the **PHY IP Core for PCI Express (PIPE)** under **PCI Express** in the Interfaces menu.

Stratix V transceivers support x1, x4, and x8 lane configurations. In a PCIe x1 configuration, the PCS and PMA blocks of each channel are clocked and reset independently. PCIe x4 and x8 configurations support channel bonding for four-lane and eight-lane PCIe links. In these bonded channel configurations, the PCS and PMA blocks of all bonded channels share common clock and reset signals.

Transceiver Datapath Configuration

The transceiver datapath configuration using the PHY IP Core for PCI Express (PIPE) configuration is shown in Figure 4–8.

Figure 4–8. PCIe Gen1 and Gen2 PIPE Datapath Configuration

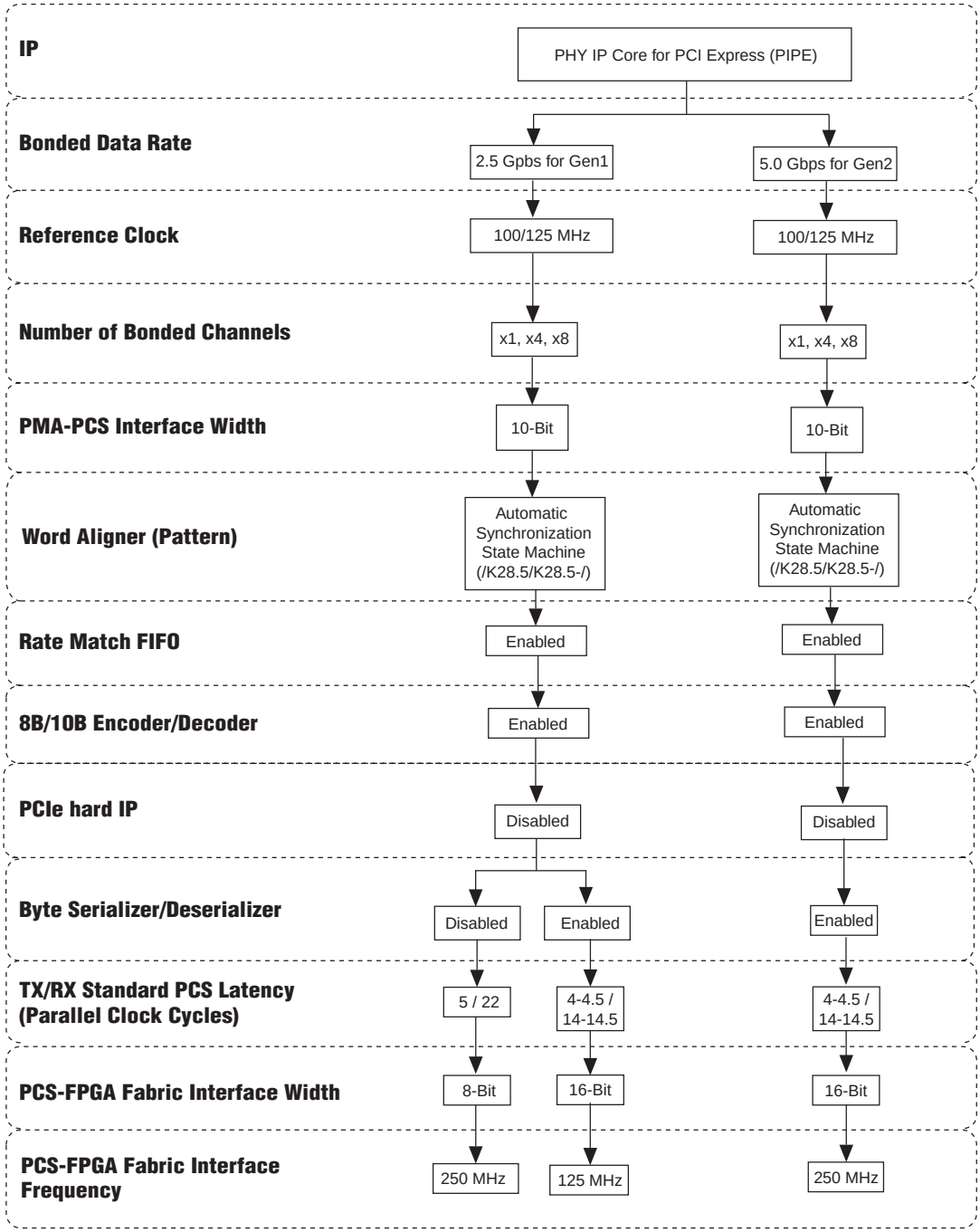
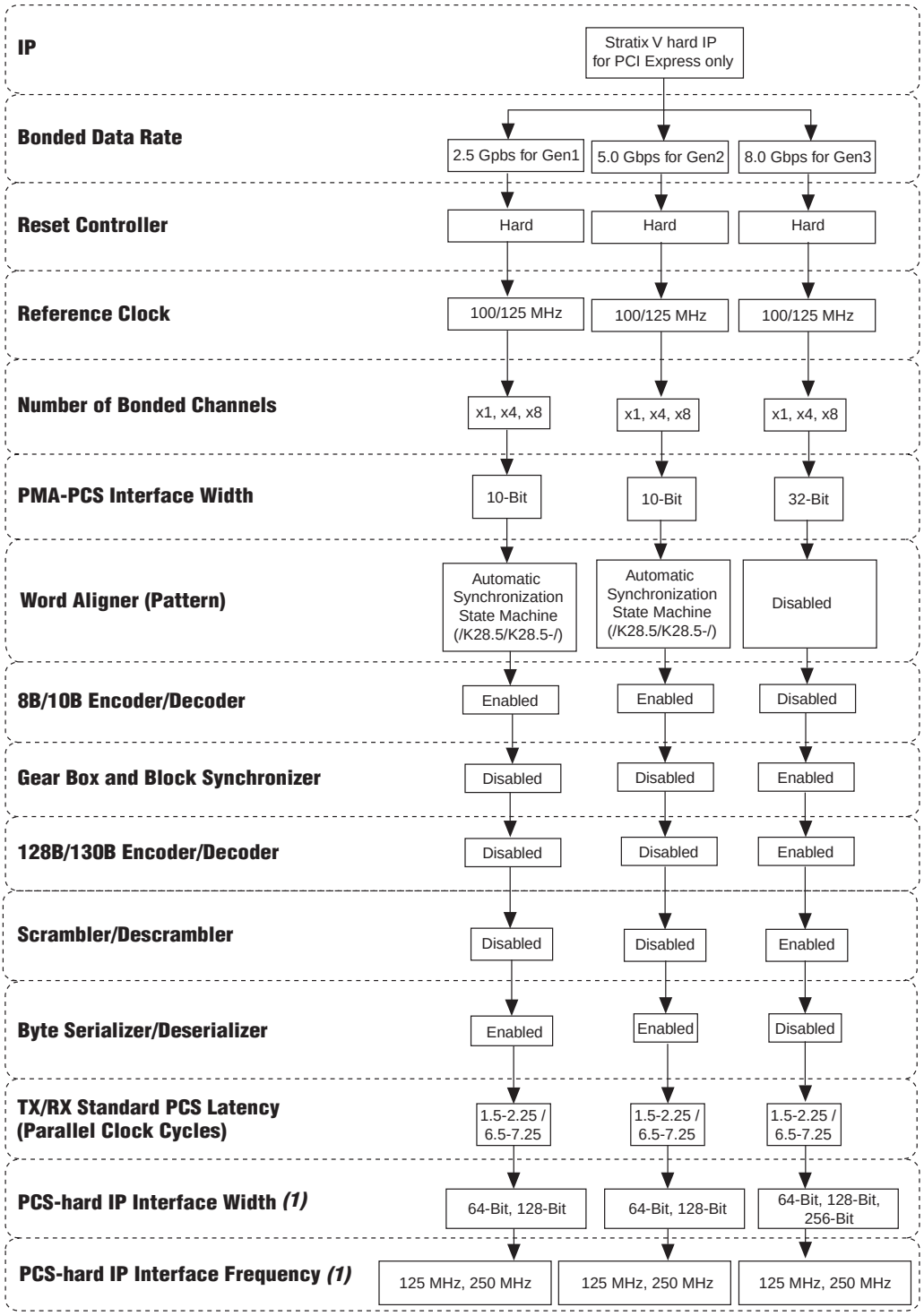


Figure 4–9 shows the transceiver configurations allowed for Gen1, Gen2, and Gen3 in a Stratix V hard IP for PCIe.

Figure 4–9. PCIe Gen1, Gen2, and Gen3 Hard IP Datapath Configuration *(1)*



Note for Figure 4–9:

(1) For more information, refer to the *Stratix V Hard IP for PCI Express User Guide*.

Transceiver Channel Datapath

Figure 4–10 shows the Stratix V transmitter and receiver channel datapath for PCIe Gen1/Gen2 configurations.

Figure 4–10. Transceiver Channel Datapath for PCIe Gen1/Gen2 Configurations

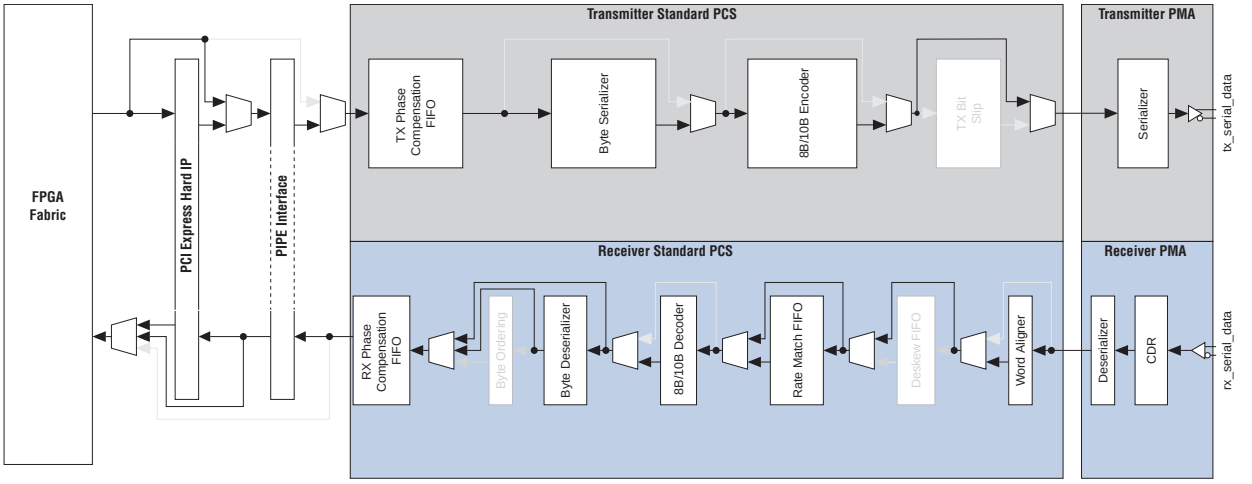
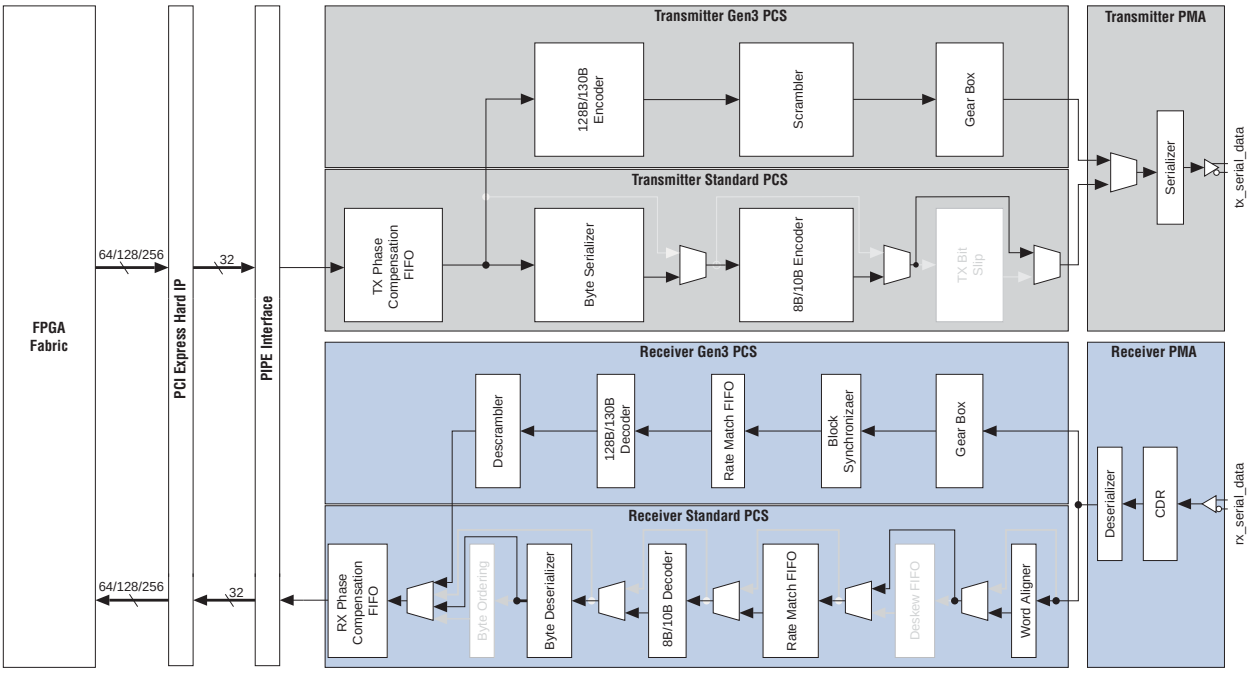


Figure 4–11 shows the Stratix V transmitter and receiver channel datapath for PCIe Gen1/Gen2/Gen3 configurations when **PCI Express Base Specification Version 3.0** is enabled.

Figure 4–11. Transceiver Channel Datapath for PCIe Gen1/Gen2/Gen3 Configurations



For more information about the blocks in the transmitter datapath, refer to the *Transceiver Architecture in Stratix V Devices* chapter.

Supported Features for PCIe Configurations

Table 4–5 lists the features supported for a PCIe configuration for the 2.5 Gbps, 5 Gbps, and 8 Gbps data rate configurations.

Table 4–5. Supported Features for PCIe Configurations

Feature	Gen1 (2.5 Gbps)	Gen2 (5 Gbps)	Gen3 (8 Gbps)
x1, x4, x8 link configurations	Yes	Yes	Yes
PCIe-compliant synchronization state machine	Yes	Yes	Yes
±300 ppm (total 600 ppm) clock rate compensation	Yes	Yes	Yes
8-bit FPGA fabric-transceiver interface	Yes	—	—
16-bit FPGA fabric-transceiver interface	Yes	Yes	—
32-bit FPGA fabric-transceiver interface	—	—	Yes
64-bit/128-bit/256-bit hard IP interface width	Yes	Yes	Yes
Transmitter buffer electrical idle	Yes	Yes	Yes
Receiver Detection	Yes	Yes	Yes
8B/10B encoder disparity control when transmitting compliance pattern	Yes	Yes	—
128B/130B encoder disparity control when transmitting compliance pattern	—	—	Yes
Power state management	Yes	Yes	Yes
Receiver status encoding	Yes	Yes	Yes
Dynamic switching between 2.5 Gbps and 5 Gbps signaling rate	—	Yes	—
Dynamic switching between 2.5 Gbps, 5 Gbps, and 8 Gbps signaling rate	—	—	Yes
Dynamic transmitter margining for differential output voltage control	—	Yes	Yes
Dynamic transmitter buffer de-emphasis of -3.5 dB and -6 dB	—	Yes	Yes
Dynamic Gen3 transceiver pre-emphasis, de-emphasis, and equalization	—	—	Yes

PIPE 2.0 Interface

In a PCIe PIPE configuration, each channel has a PIPE interface block that transfers data, control, and status signals between the PHY-MAC layer and the transceiver channel PCS and PMA blocks. The PIPE configuration complies with the PIPE 2.0 specification. If you use a PIPE configuration, you must implement the PHY-MAC layer using soft IP in the FPGA fabric.

Besides transferring data, control, and status signals between the PHY-MAC layer and the transceiver, the PIPE interface block implements the following functions required in a PCIe-compliant physical layer device:

- Forcing the transmitter buffer in the electrical idle state
- Initiating the receiver detect sequence
- Controlling the 8B/10B encoder disparity when transmitting compliance pattern
- Managing the PCIe power states
- Indicating the completion of various PHY functions

- Encoding the receiver status and error conditions on the pipe_rxstatus[2:0] signal, conforming to the PCIe specification

Transceiver datapath clocking varies between non-bonded (x1) and bonded (x4 and x8) configurations.

For more information about transceiver datapath clocking in different PIPE configurations, refer to "Transceiver Clocking for PCIe Gen1 and Gen2" on page 4–26.

Dynamic Switching Between Gen1 (2.5 Gbps) and Gen2 (5 Gbps) Signal Rates

In a PIPE configuration, the PIPE MegaWizard Plug-In Manager provides an input signal (pipe_rate) that is functionally equivalent to the RATE signal specified in the PCIe specification. A low-to-high transition on this input signal (pipe_rate) initiates a data rate switch from Gen1 to Gen2. A high-to-low transition on the input signal initiates a data rate switch from Gen2 to Gen1. The signaling rate switch between Gen1 and Gen2 is achieved by changing the transceiver datapath clock frequency between 250 MHz and 500 MHz, while maintaining a constant, 16-bit width transceiver interface.

For more information about using the pipe_rate input signal and a timing diagram showing the sequence of rate switch events and status signals, refer to the *PHY IP Core for PCI Express (PIPE)* chapter in the *Altera Transceiver PHY IP Core User Guide*. For the power state requirements when switching between Gen1 and Gen2 data rates, refer to the PCIe Base Specification 2.0.

Transmitter Electrical Idle Generation

The PIPE interface block in Stratix V devices puts the transmitter buffer in the channel in an electrical idle state when the electrical idle input signal is asserted. During electrical idle, the transmitter buffer differential and common configuration output voltage levels are compliant to the PCIe Base Specification 2.0 for both PCIe Gen1 and Gen2 data rates.

The PCIe specification requires the transmitter buffer to be in electrical idle in certain power states. For more information about input signal levels required in different power states, refer to "Power State Management".

For more information about the electrical idle input signal and transmitter buffer state, refer to the *PHY IP Core for PCI Express (PIPE)* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Power State Management

The PCIe specification defines four power states—P0, P0s, P1, and P2—that the physical layer device must support to minimize power consumption:

- P0 is the normal operating state during which packet data is transferred on the PCIe link.
- P0s, P1, and P2 are low-power states into which the physical layer must transition as directed by the PHY-MAC layer to minimize power consumption.

The PIPE interface in Stratix V transceivers provides an input port for each transceiver channel configured in a PIPE configuration.

For more information about input and status signals to manipulate power states, refer to the *PHY IP Core for PCI Express (PIPE)* chapter in the *Altera Transceiver PHY IP Core User Guide*.

When transitioning from the P0 power state to lower power states (P0s, P1, and P2), the PCIe specification requires the physical layer device to implement power saving measures. Stratix V transceivers do not implement these power saving measures except for putting the transmitter buffer in electrical idle in the lower power states.

8B/10B Encoder Usage for Compliance Pattern Transmission Support

The PCIe transmitter transmits a compliance pattern when the Link Training and Status State Machine (LTSSM) enters the Polling.Compliance substate. The Polling.Compliance substate is used to assess if the transmitter is electrically compliant with the PCIe voltage and timing specifications.

For more information about the 8B/10B signals required for compliance pattern transmission support, refer to the *PHY IP Core for PCI Express (PIPE)* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Receiver Electrical Idle Inference

The PCIe protocol allows inferring the electrical idle condition at the receiver instead of detecting the electrical idle condition with analog circuitry.

In all PIPE configurations, (x1, x4, and x8), each receiver channel PCS has an optional Electrical Idle Inference module that implements the electrical idle inference conditions specified in the PCIe Base Specification 2.0.

Receiver Status

The PCIe specification requires the PHY to encode the receiver status on a 3-bit status signal (`pipe_rxstatus[2:0]`). This status signal is used by the PHY-MAC layer for its operation. The PIPE interface block receives status signals from the transceiver channel PCS and PMA blocks, and encodes the status on the `pipe_rxstatus[2:0]` signal to the FPGA fabric. The encoding of the status signals on the `pipe_rxstatus[2:0]` signal conforms to the PCIe specification.

For information about the encoding of status signals on the `pipe_rxstatus[2:0]` signal, refer to the *PHY IP Core for PCI Express (PIPE)* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Receiver Detection

The PIPE interface block in Stratix V transceivers provides an input signal (`pipe_txdetectrx_loopback`) for the receiver detect operation required by the PCIe protocol during the Detect state of the LTSSM. When the `pipe_txdetectrx_loopback` signal is asserted in the P1 power state, the PCIe interface block sends a command signal to the transmitter buffer in that channel to initiate a receiver detect sequence. In the P1 power state, the transmitter buffer must always be in the electrical idle state. After receiving this command signal, the receiver detect circuitry creates a step voltage at the output of the transmitter buffer. If an active receiver (that complies with the PCIe input impedance requirements) is present at the far end, the time constant of

the step voltage on the trace is higher when compared with the time constant of the step voltage when the receiver is not present. The receiver detect circuitry monitors the time constant of the step signal seen on the trace to determine if a receiver was detected. The receiver detect circuitry requires a 125-MHz clock for operation that you must drive on the `fixedclk` port.

For the receiver detect circuitry to function reliably, the transceiver on-chip termination must be used and the AC-coupling capacitor on the serial link and the receiver termination values used in your system must be compliant with the PCIe Base Specification 2.0.

The PIPE core provides a 1-bit PHY status (`pipe_phystatus`) and a 3-bit receiver status signal (`pipe_rxstatus[2:0]`) to indicate whether a receiver was detected or not, as per the PIPE 2.0 specifications.

For more information about input signals and status signals related to receiver detection, refer to the *PHY IP Core for PCI Express (PIPE)* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Clock Rate Compensation Up to ±300 PPM

In compliance with the PCIe protocol, Stratix V receiver channels have a rate match FIFO to compensate for small clock frequency differences up to ±300 ppm between the upstream transmitter and the local receiver clocks.

For more information about the rate match FIFO operation in a PCIe configuration, refer to the *Transceiver Architecture in Stratix V Devices* chapter.

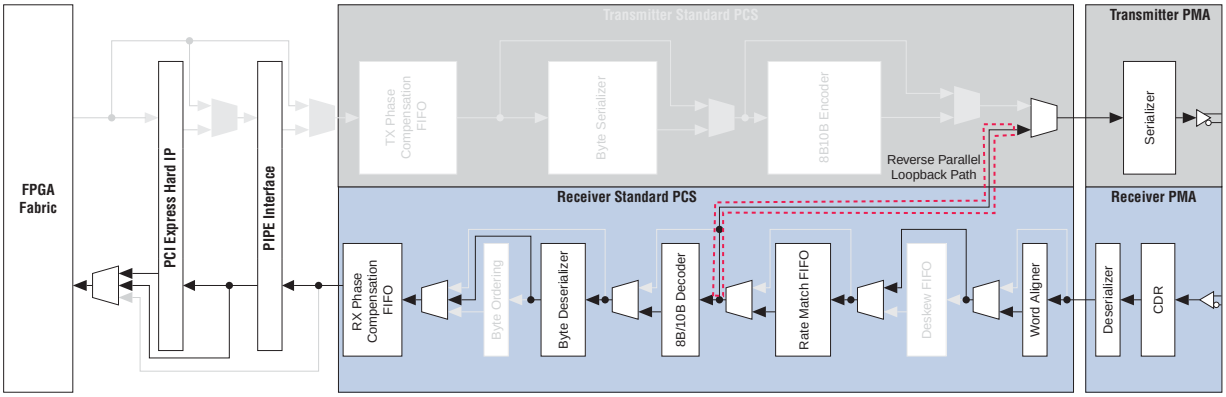
For more information about status signals and registers for the rate match FIFO, refer to the *PHY IP Core for PCI Express (PIPE)* chapter in the *Altera Transceiver PHY IP Core User Guide*.

PCIe Reverse Parallel Loopback

PCIe reverse parallel loopback is only available in a PCIe functional configuration for Gen1 and Gen2 data rates. As shown in Figure 4–12, the received serial data passes through the receiver CDR, deserializer, word aligner, and rate matching FIFO buffer. The data is then looped back to the transmitter serializer and transmitted out through the transmitter buffer. The received data is also available to the FPGA fabric through the port. This loopback mode is compliant with the PCIe specification 2.0. Stratix V devices provide an input signal to enable this loopback mode.

This is the only loopback option supported in PIPE configurations.

Figure 4–12. PCIe Reverse Parallel Loopback Mode Datapath (Grayed-out Blocks are Inactive)



Supported Features for PCIe Gen3

The PCIe Gen3 hard IP PCS supports the Gen3 base specification. PCIe Gen3 operation is configured using the Stratix V hard IP for PCI Express IP. Enabling **PCI Express Base Specification Version 3.0** selects a 32-bit wide PIPE interface. Enabling **PCI Express Base Specification Version 2.1** selects the 16-bit/8-bit wide PIPE interface and limits operation to PCIe Gen2 and Gen1 data rates.

Block Synchronization (Word Aligner)

The block synchronizer aligns the recovered serial data coming from the CDR to 130-bit word boundaries. The block synchronizer delineates the word boundaries by searching and identifying the Electrical IDLE Exit Sequence Ordered Set (EIEOS) or the Last FTS OS and SKP ordered set to correctly identify the word boundary from the incoming serial data stream. The block synchronizer continues to realign to a new block boundary following the receipt of an SKP ordered set because of varying word lengths.

Rate Match/Clock Compensation FIFO

To accommodate PCIe protocol requirements and to compensate for clock frequency differences of up to ±300 ppm between source and termination equipment, Stratix V receiver channels have a rate match FIFO. The rate match FIFO adds or deletes four SKP characters (32 bits) to keep the FIFO from becoming empty or full. It monitors the block synchronizer for a `skip_found` signal. If the rate match FIFO is almost full, the FIFO deletes four SKP characters. If the rate match FIFO is nearly empty, the FIFO inserts an SKP character at the start of the next available SKP ordered set.

Encoder/Decoder

Unlike PCIe Gen1 and Gen2, the PCIe Gen3 encoder/decoder does not use 8B/10B encoding. The Stratix V PCIe Gen3 encoder/decoder uses a 2-bit sync header and a 128-bit data word. The PCS encoder appends the two sync header bits to every 128 bits of data and enables scrambling for the data packets except for ordered set packets and the first symbol of a TS1/TS2 ordered set. The encoder/decoder

continuously enables or disables scrambling, based on whether the payload being processed is an ordered set or a data packet. If an Electrical IDLE Exit Ordered Set or a Fast Training Sequence Ordered Set is received, the scrambler is reset to the initial seed value. The encoder/decoder also monitors the data stream for ordered set and sync header bit violations.

Gen3 Gear Box

The PCIe 3.0 base specification requires a block size of 130 bits with the exception of SKP ordered sets, which can be 66, 98, 130, 162, or 194 bits in length. The 130-bit block of data generated by the 128B/130B encoder and variable length SKP characters must be reordered in 32-bit parallel data segments that the PMA serializer can accept. Stratix V devices employ a gear box to accommodate this fractional bit difference between the 130-bit data word and a fixed 32-bit serialization PMA factor for Gen3.

Scrambler/Descrambler

Scrambling and descrambling are used during PCIe Gen3 operation to guarantee adequate transitions for the receiver in order to correctly regenerate the recovered clock. The 2-bit sync header bit, ordered set, and the first symbol of the TS1/TS2 ordered set are never scrambled.

PIPE Gen3 Interface

PCIe Gen3 is a new feature added to Stratix V devices. The Stratix V PCS supports PCI Express 3.0 base specification. The PIPE interface has been expanded to a 32-bit wide interface. The PIPE interface controls PHY functions such as transmission of electrical idle, receiver detection, and speed negotiation and control. In summary, the Gen3 PIPE interface block performs the following:

- Dynamic clock selection between Gen1, Gen2, and Gen3 speeds
- Gen3 auto speed negotiation (ASN)
- Gen3 Electrical Idle Entry and Exit detections/CDR Control
- Dynamic Gen3 and Gen2/Gen1 PCS data rate switching
- Dynamic transceiver PMA data rate and PLL switching

Auto-Speed Negotiation Block

PCIe Gen3 mode enables ASN (auto-speed negotiation) between Gen1 (2.5 Gbps), Gen2 (5.0 Gbps), and Gen3 (8.0 Gbps) signaling data rates. The signaling rate switch is accomplished through frequency scaling and configuration of the PMA and PCS blocks using a fixed 32-bit wide PIPE Interface.

The PMA switches clocks between Gen1, Gen2, and Gen3 data rates in a glitch-free manner. For a non-bonded x1 channel, an ASN module facilitates speed negotiation in that channel. For bonded x4 and x8 channels, the ASN module selects the master channel to control the rate switch. The master channel distributes the speed change request to the other PMA and PCS channels.

Table 4–6. PIPE Gen3 32-Bit PCS Clock Rates

PCIe Gen3 Capability Mode Enabled	Gen1	Gen2	Gen3
Lane data rate	2.5G	5G	8G
PCS clock frequency	250 MHz	500 MHz	250 MHz
FPGA Core IP clock frequency	62.5 MHz	125 MHz	250 MHz
PIPE interface width	32-bit	32-bit	32-bit
Rate[1:0]	00	01	10

The PCIe Gen3 speed negotiation process is initiated by writing a 1 to bit 5 of the Link Control register of the root port, causing a PIPE rate signal change from the hard IP. The ASN then places the PCS in reset, dynamically shuts down the clock paths to disengage the current active state PCS (either Standard PCS or Gen3 PCS). If a switch to or from Gen3 is requested, the ASN automatically selects the correct PCS clock paths and datapath selection in the multiplexers. The ASN block then sends a request to the PMA block to switch the data rate change and waits for a rate change done signal for confirmation. When the PMA completes the rate change and sends confirmation to the ASN block, ASN enables the clock paths to engage the new PCS block and releases the PCS reset. Successful completion of this process is indicated by assertion of the `pipe_phystatus` signal by the ASN block to the hard IP block.

Transmitter Electrical IDLE Generation

The PIPE Gen3 interface under the control of the hard IP block may place the transmitter in electrical idle during low power states and the ASN process. Before the transmitter enters electrical idle, the HIP sends an electrical idle order set (EIOS) to the PHY. For Gen1 and Gen2, the order set format is COM, IDL, IDL, IDL. For Gen3, the order set format consists of 16 symbols with value 0x66.

During electrical idle, the transmitter buffer differential and common mode voltage levels are compliant to the PCIe Base Specification 3.0.

For more information about the electrical idle input signal and transmitter buffer state, refer to "Gen3 Power State Management".

Receiver Electrical IDLE Inference

If there is no activity on the link for a period of time, the Inferring Electrical Idle condition is detected by the receiver PHY. These conditions are specified according to Table 4-11 of the PCI Express Base Specification, Rev 3.0.

Gen3 Power State Management

The PCIe base specification defines low power states for PHY layer devices to minimize power consumption. Stratix V Gen3 PCS does not implement these power saving measures, except when placing the transmitter buffer in electrical idle state in the low power states. In P2 low power state, Stratix V devices do not disable the PIPE block clock.

CDR Control Block

The CDR control block controls the PMA CDR to obtain bit and symbol alignment and deskew within the allocated time, and generates status signals for other PCS blocks. The PCIe base specification requires that the receiver L0s power state exit time be a maximum of 4 µs for Gen1, 2 µs for Gen2, and 4 µs for Gen3 signaling rates. Stratix V devices have an improved CDR control block to accommodate fast lock times when the CDR must relock to the new multiplier/divider settings when entering or exiting Gen3 speeds.

Transceiver Clocking and Channel Placement Guidelines

This section describes the transceiver clocking and channel placement guidelines for Gen1 and Gen2 PIPE configurations.

Transceiver Clocking for PCIe Gen1 and Gen2

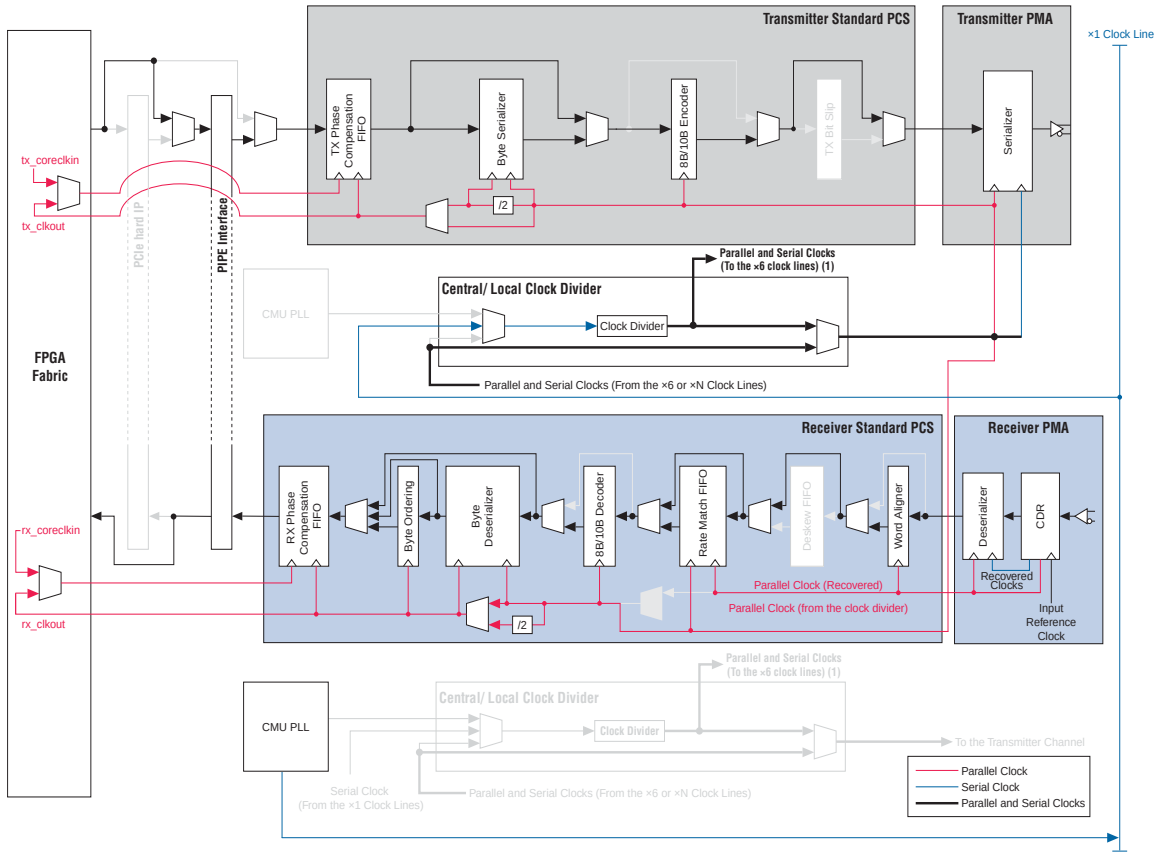
This section describes the transceiver clocking configurations for PIPE configurations.

PIPE x1 Configuration

Figure 4–13 shows the transceiver clocking configuration in a PIPE x1 configuration.

The high-speed serial clock is provided by the CMU PLL in a channel different from that of the data channel. The local clock divider block in the data channel generates a parallel clock from this high-speed clock and distributes both clocks to the PMA and PCS of the data channel.

Figure 4–13. Transceiver Clocking in a Gen1/Gen2 PIPE x1 Configuration



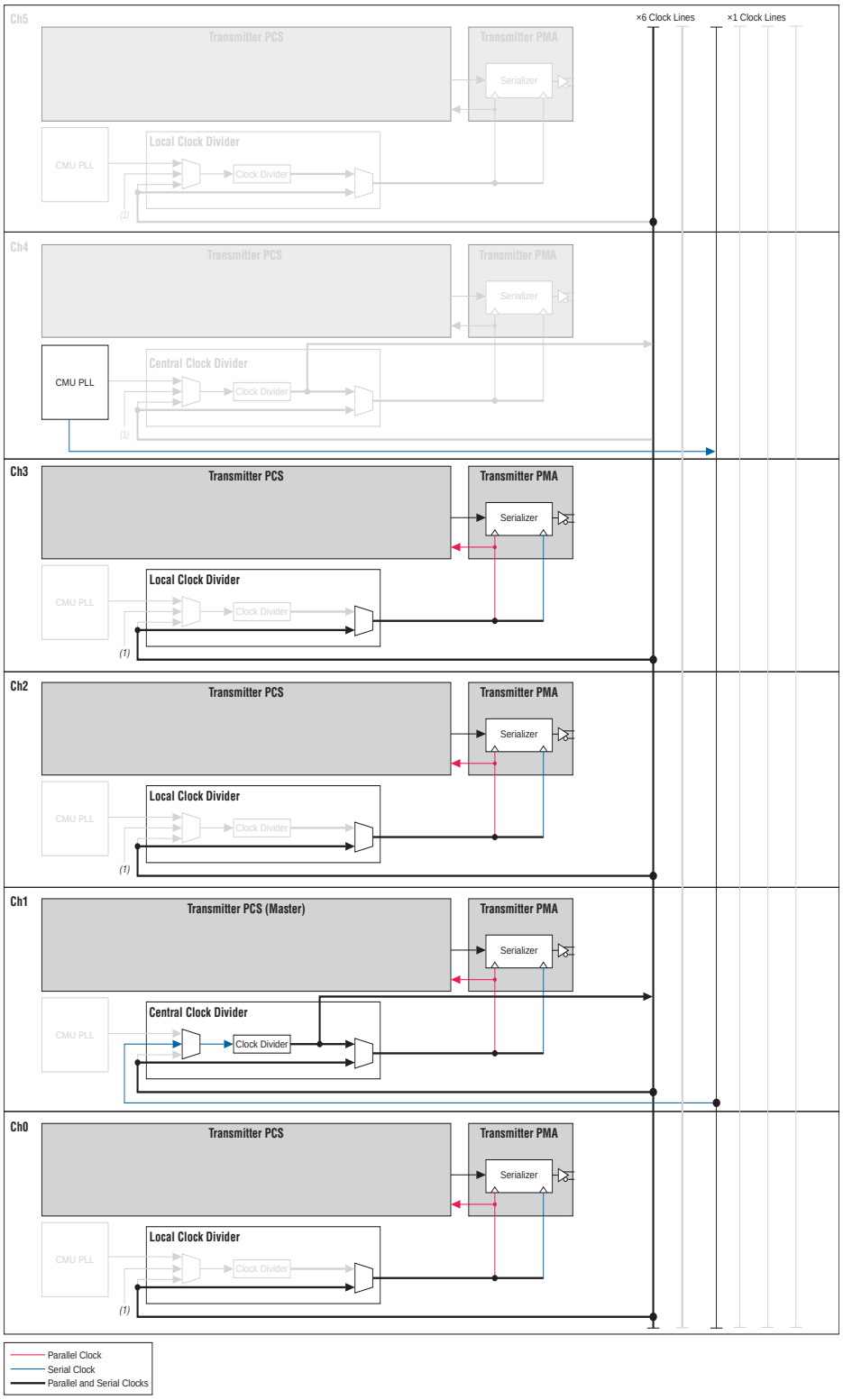
Note to Figure 4–13:

(1) Only available in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

PIPE x4 Configuration

Figure 4–14 shows transmitter clocking for a PIPE x4 bonded configuration. Clocking within the PCS is independent for each receiver channel. Clocking is bonded only for transmitter channels, while the control signals are bonded for both transmitter and receiver channels. The Quartus II software automatically places the transmit CMU PLL and master channel in either channel 1 or channel 4 within a transceiver bank, as shown in Figure 4–14.

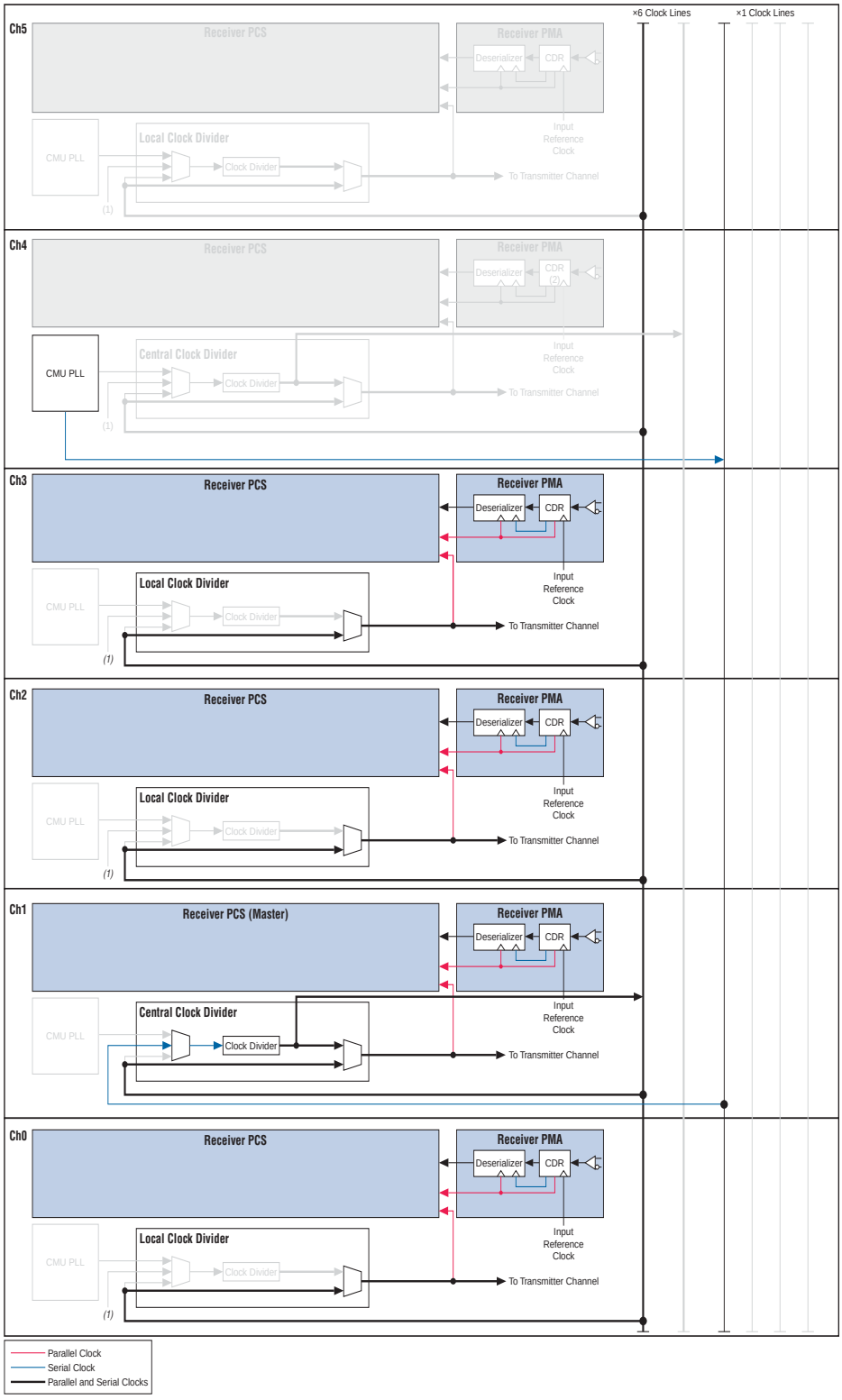
Figure 4–14. Transmitter Clocking in a Gen1/Gen2 PIPE x4 Configuration



Note to Figure 4–14:

(1) Serial clock and parallel clock from the x6 clock lines.

Figure 4–15. Receiver Clocking in a Gen1/Gen2 PIPE x4 Configuration



Note to Figure 4–15:

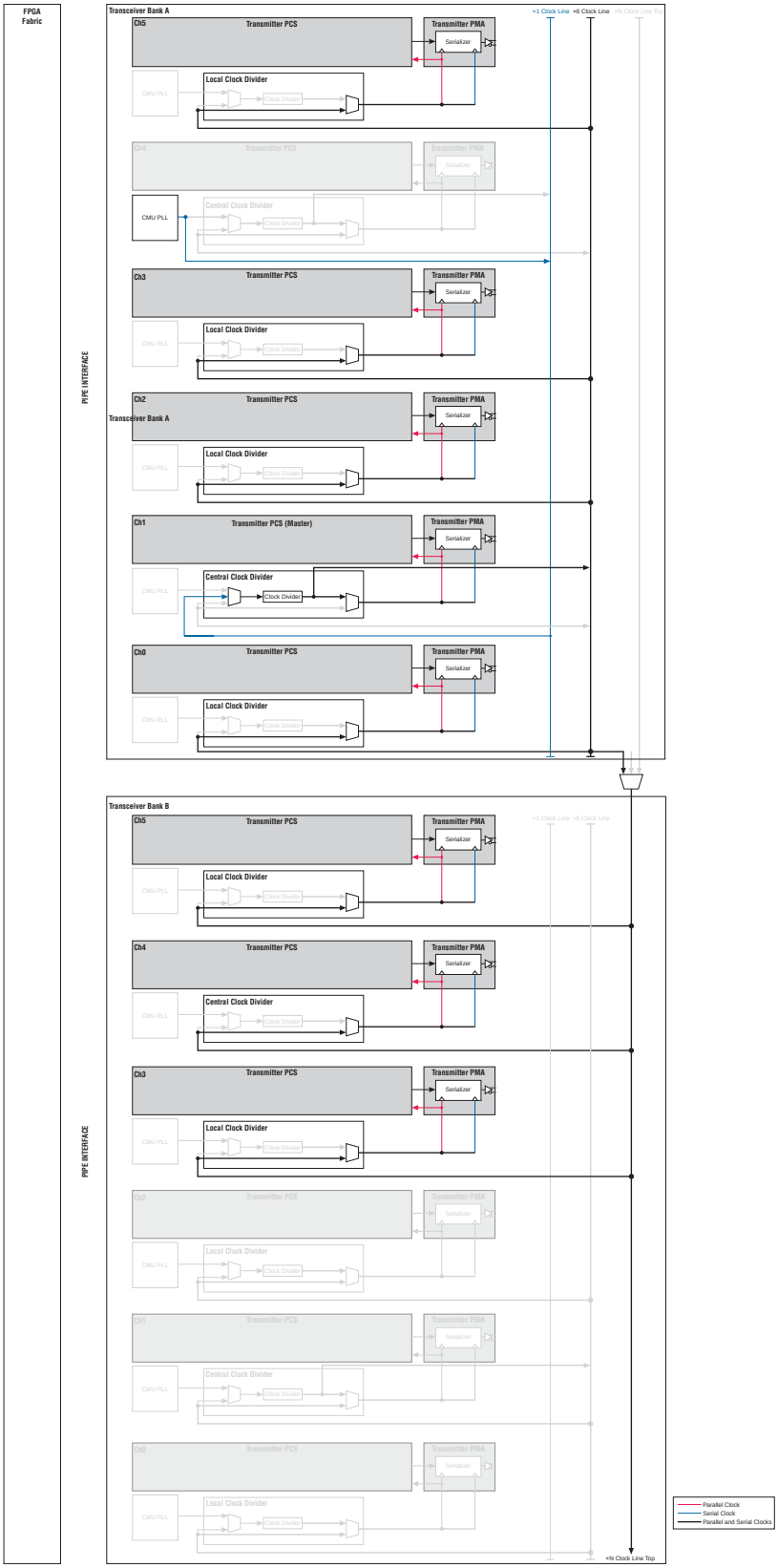
(1) Serial clock and parallel clock from the x6 clock lines.

PIPE x8 Configuration

Figure 4–16 shows clocking for PMA and PCS blocks in the x8 PCIe bonded configuration. Clocking is independent for receiver channels. Clocking and control signals are bonded only for transmitter channels.

For more information about clocking in Stratix V devices, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

Figure 4–16. Transceiver Clocking in a Gen1/Gen2 PIPE x8 Configuration



Transceiver Channel Placement Guidelines for Gen1 and Gen2 PIPE Configurations

Table 4–7 lists the physical placement of PIPE channels in x1, x4, and x8 bonding configurations. The Quartus® II software automatically places the CMU PLL in a channel different from that of the data channels.

Table 4–7. Gen2 and Gen1 PIPE Configuration Channel Placement

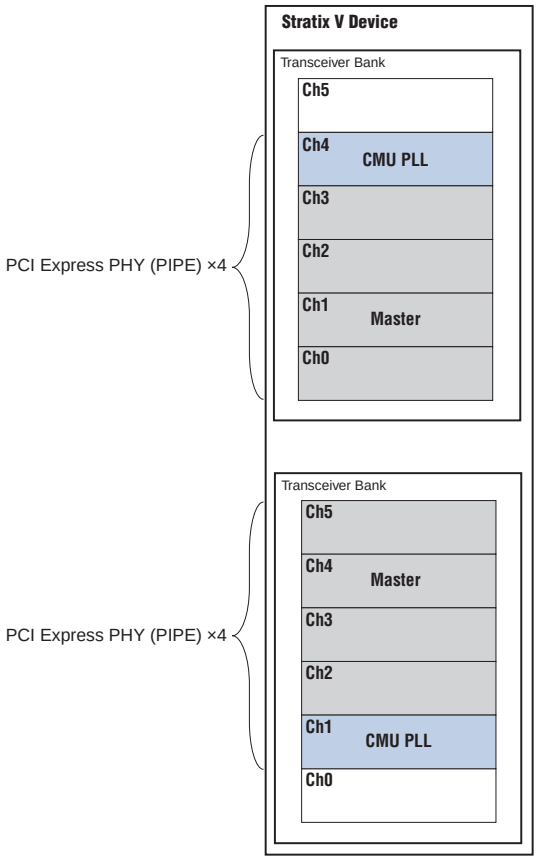
Configuration	Data Channel Placement	Channel Utilization Using CMU PLL (1)	Channel Utilization Using ATX PLL (1)
x1	Any channel	2	1
x4	Contiguous channels	5	4
x8	Contiguous channels	9	9

Note to Table 4–7:

(1) Placement by the Quartus II software may vary with design, thus resulting in higher channel usage.

For PIPE x1 configurations, the channel can be placed anywhere within a transceiver bank that contains the transmitter PLL. Figure 4–17 and Figure 4–18 show examples of channel placement for PIPE x4 and x8 configurations. In a PIPE x4 configuration, the four channels must be contiguous and within the same transceiver bank, but they can be placed in any order. The CMU PLL must be within the same transceiver bank.

Figure 4–17. Example of PIPE x4 Channel Placement Using the CMU PLL *(1), (2), (3)*

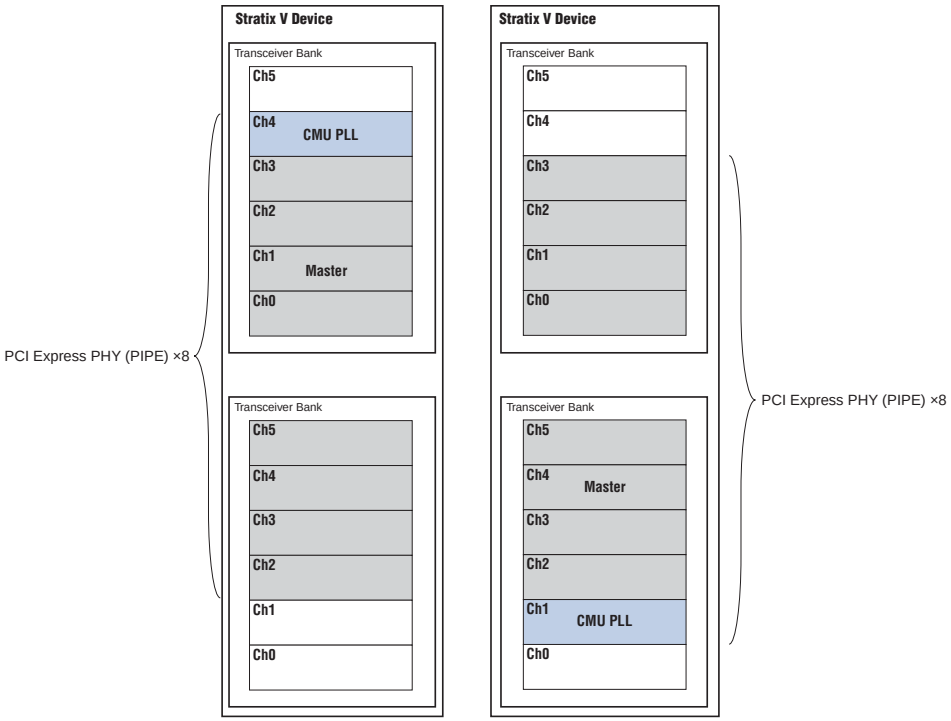


Notes to Figure 4–17:

(1) Channels shaded in blue provide the transmit CMU PLL generating the high-speed serial clock.
(2) Channels shaded in gray are data channels.
(3) The Quartus II software automatically selects the CMU PLL in either channel 1 or channel 4 within a transceiver bank.

In a PIPE x8 configuration, the eight channels must be contiguous, but they can be placed in any order.

Figure 4–18. Example of PIPE x8 Channel Placement Using a CMU PLL *(1), (2)*



Notes to Figure 4–18:

(1) Channels shaded in blue provide the transmit CMU PLL generating the high-speed serial clock.

(2) Channels shaded in gray are data channels.

Transceiver Clocking for PCIe Gen3 (Hard IP)

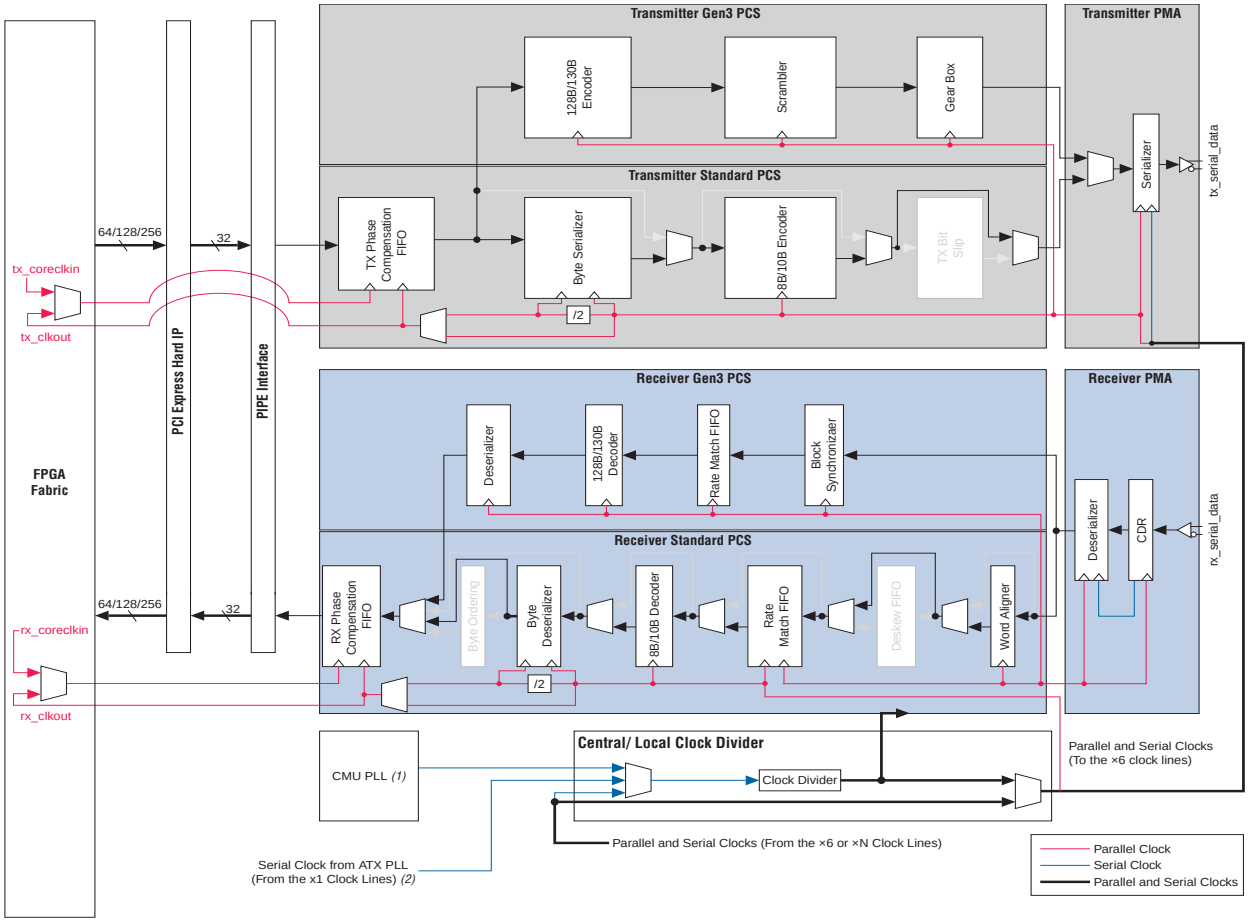
This section describes the transceiver clocking topology for the PCIe Gen3 configuration. In a PCIe x1, x4, and x8 Gen3 Mode, both a channel PLL (CMU PLL) from transceiver physical channel 1 or 4 of the transceiver bank and an ATX PLL are used to generate the high-speed serial clock and support ASN. The CMU PLL supports Gen1 and Gen2 data rates while the ATX PLL supports Gen3 data rates. To enable rapid switching between Gen1, Gen2, and Gen3 data rates, a multiplexer selects either the free running CMU PLL for Gen1 and Gen2 data rates or the free running ATX PLL for Gen3 data rates. PLL reconfiguration is not used to support ASN.

PCIe Gen3 is only supported when you select the Stratix V hard IP for PCI Express option in the MegaWizard Plug-In Manager. For channel placement guidelines for hard IP mode, refer to "Channel Placement Gen3" in the *Stratix V Hard IP for PCI Express User Guide*.

Hard IP x1 Configuration

Figure 4–19 shows the transceiver clocking topology for PCIe x1 Gen3 configuration using the Stratix V hard IP for PCI Express.

Figure 4–19. Transceiver Clocking in a Gen1/Gen2/Gen3 PCIe x1 Configuration



Notes to Figure 4–19:

(1) For Gen1 and Gen2, use the CMU PLL.

(2) For Gen3, use the ATX PLL.

For **PCIe x1 Gen3** using hard IP, the CMU PLL (transceiver physical channel 1) and the bottom ATX PLL of the transceiver bank are configured to generate the high-speed serial clock for the transmitter datapath clock and the rate matcher side of the FIFO in the receiver datapath if rate matching is enabled for the data channel. Two transceiver channels are needed to implement PCIe x1 Gen3, one for the data channel and one for the CMU PLL. The local clock divider block in the data channel generates a parallel clock from this high-speed serial clock and distributes both clocks to the PMA and PCS of the data channel.

For additional information about channel placement implementation for PCIe Gen3 x1, refer to "Channel Placement Gen3" in the *Stratix V Hard IP for PCI Express User Guide*.

Hard IP x4 Configuration

Figure 4–20 shows the transceiver clocking topology for the transmitter PCS with the **PCIe x4 Gen 3** enabled using the hard IP.

Figure 4–20. Transmitter Clocking in a Gen1/Gen2/Gen3 PCIe x4 Configuration

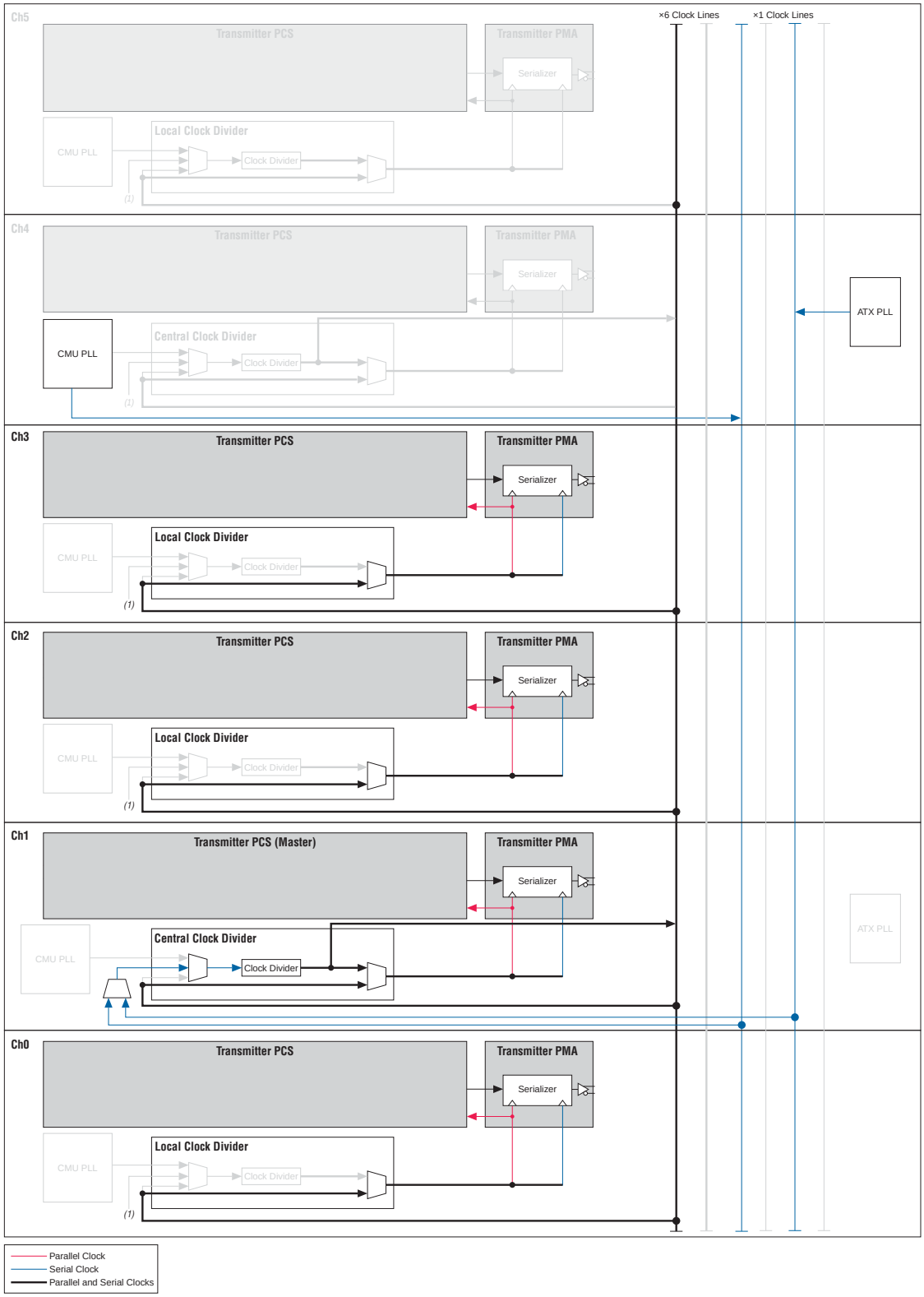
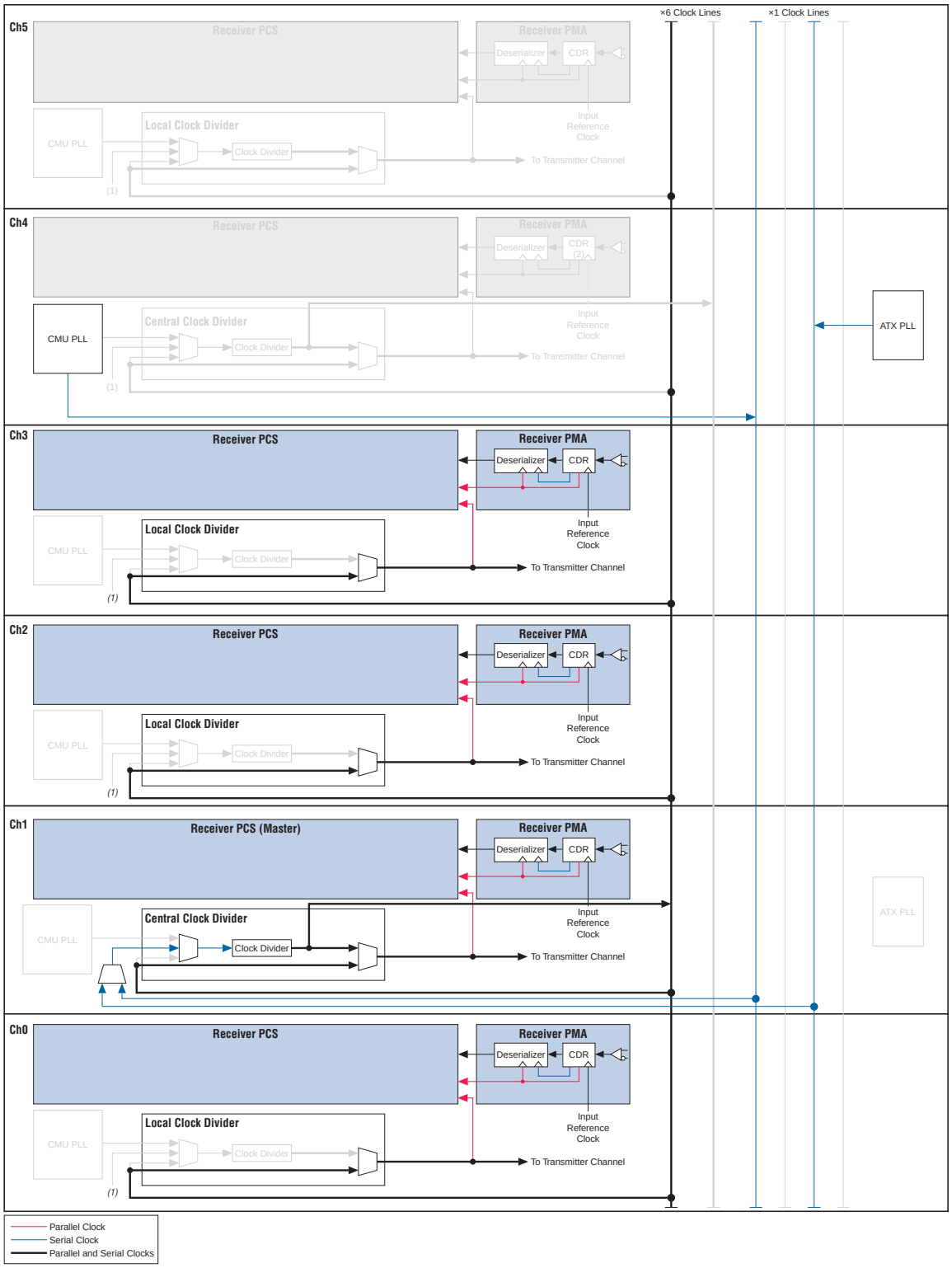


Figure 4–21 shows the transceiver clocking topology for the receiver PCS with the **PCIe x4 Gen 3** enabled using the hard IP.

Figure 4–21. Receiver Clocking in a Gen1/Gen2/Gen3 PCIe x4 Configuration



In a **PCIe x4 Gen3** using hard IP, the CMU PLL (transceiver physical channel 4) and the top ATX PLL of the transceiver bank are configured to generate the high-speed serial clock. A total of five transceiver channels are required to implement PCIe x4 Gen3, including four data channels and one channel for the CMU PLL. The Quartus II software automatically selects channel 1 in the transceiver bank as the master channel. Channel 1 bonds and drives all the transmitter datapath's clocking and the rate matcher side of the FIFO in the receiver datapaths if rate matching is enabled for the four data channels. The local clock divider blocks in each data channel generates the parallel clock from the high-speed serial clock and distributes both clocks to the PMA and PCS of that data channel.

For additional information about channel placement implementation for PCIe Gen3 x4, refer to "Channel Placement Gen3" in the *Stratix V Hard IP for PCI Express User Guide*.

Hard IP x8 Configuration

In a **PCIe x8 Gen3** using hard IP, the CMU PLL (transceiver physical channel 4) and the top or bottom ATX PLL of the lower transceiver bank with the PCIe Gen3 hard IP are configured to generate the high-speed serial clock. A total of nine transceiver channels are required to implement PCIe x8 Gen3, including eight data channels and one channel for the CMU PLL. The Quartus II software automatically selects channel 4 in the transceiver bank as the master channel. Channel 4 bonds and drives all the transmitter datapath's clocking and the rate matcher side of the FIFO in the receiver datapaths if rate matching is enabled for the eight data channels. The local clock divider blocks in each data channel generates the parallel clock from this high-speed serial clock and distributes both clocks to the PMA and PCS of that data channel. The master channel in the x8 case is not a data channel.

For additional information about channel placement implementation for PCIe Gen3 x8, refer to "Channel Placement Gen3" in the *Stratix V Hard IP for PCI Express User Guide*.

CPRI and OBSAI—Deterministic Latency Protocols

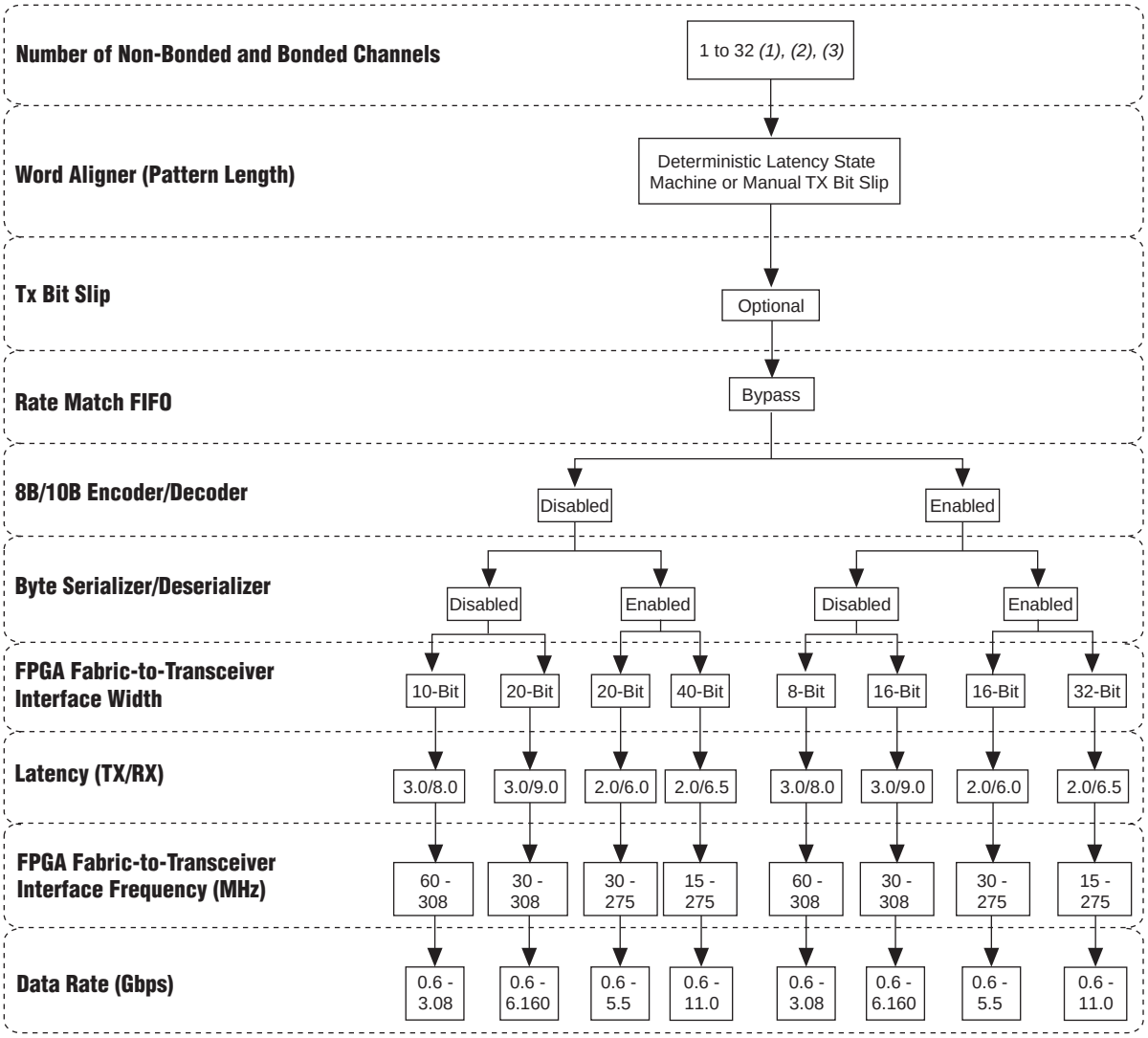
Stratix V devices have a deterministic latency option available for use in high-speed serial interfaces such as the Common Public Radio Interface (CPRI) and OBSAI Reference Point 3 (OBSAI RP3). Both CPRI and OBSAI RP3 protocols place stringent requirements on the amount of latency variation that is permissible through a link that implements these protocols.

Transceiver Datapath Configuration

Figure 4–22 shows the available options for the deterministic latency datapath configuration. The maximum frequencies shown in Figure 4–22 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–22. Deterministic Latency Datapath Configuration



Notes to Figure 4–22:

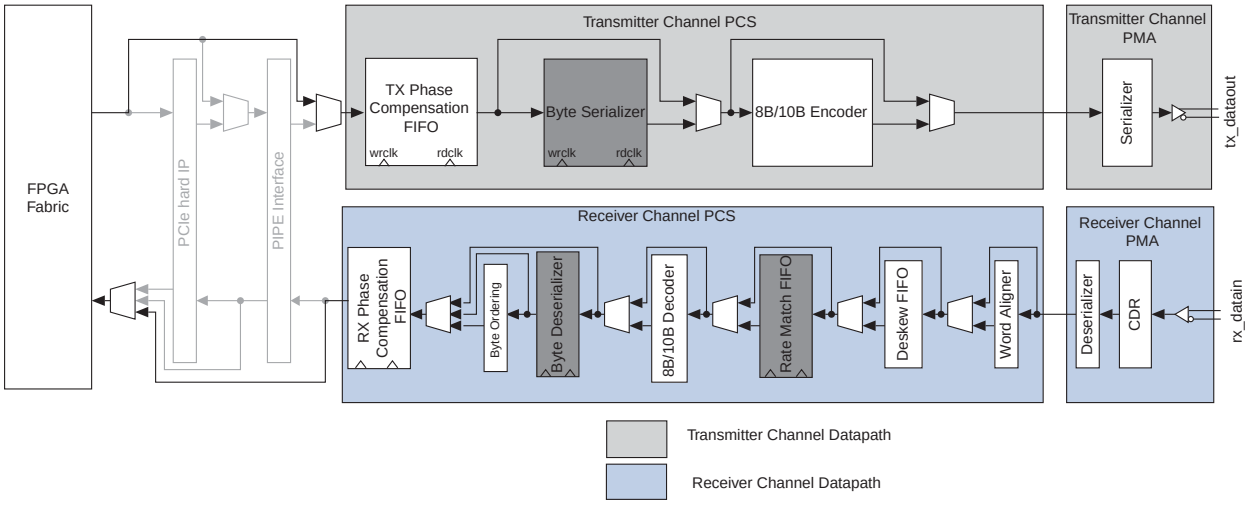
(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) The TX-client feedback path to the transmit PLL is only supported in a non-bonded single lane instance.

Figure 4–23 shows the transceiver datapath when using deterministic latency mode.

Figure 4–23. Transceiver Datapath in Deterministic Latency Mode



Receiver Phase Compensation FIFO in Register Mode

To remove the latency uncertainty through the receiver's phase compensation FIFO, the receiver and transmitter phase compensation FIFOs are always set to register mode. In register mode, the phase compensation FIFO acts as a register and thereby removes the uncertainty in latency. The latency through the phase compensation FIFO in register mode is one clock cycle.

The following options are available:

- 8/10-bit-width mode with one of the following:
  - 8-bit channel width and 8B/10B encoder enabled
  - 10-bit channel width with 8B/10B disabled
- 16/20-bit-width mode with one of the following:
  - 16-bit channel width and 8B/10B encoder enabled
  - 20-bit channel width with 8B/10B disabled

Transmitter Phase Compensation FIFO in Register Mode

In register mode, the phase compensation FIFO acts as a register and thereby removes the uncertainty in latency. The latency through the transmitter and receiver phase compensation FIFO in register mode is one clock cycle.

Channel PLL Feedback

To implement the deterministic latency functional mode, the phase relationship between the low-speed parallel clock and channel PLL input reference clock must be deterministic. A feedback path is enabled to ensure a deterministic relationship between the low-speed parallel clock and channel PLL input reference clock.

To achieve deterministic latency through the transceiver, the reference clock to the channel PLL must be the same as the low-speed parallel clock. For example, if you need to implement a data rate of 1.2288 Gbps for the CPRI protocol, which places stringent requirements on the amount of latency variation, you must choose a reference clock of 122.88 MHz to allow the usage of a feedback path from the channel PLL. This feedback path reduces the variations in latency.

When you select this option, provide an input reference clock to the channel PLL that is of the same frequency as the low-speed parallel clock.

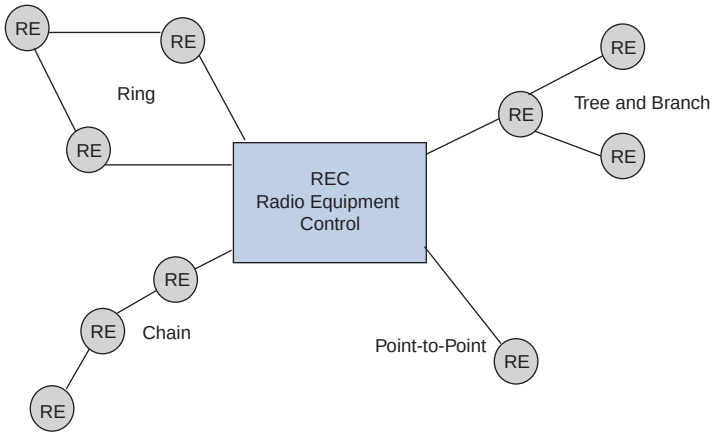
CPRI and OBSAI

You can use the deterministic latency functional mode to implement protocols such as CPRI and OBSAI.

The CPRI interface defines a digital point-to-point interface between the Radio Equipment Control (REC) and the Radio Equipment (RE), allowing flexibility in either co-locating the REC and the RE, or a remote location of the RE.

Figure 4–24 shows various CPRI topologies. In most cases, CPRI links are between REC and RE modules or between two RE modules in a chain configuration.

Figure 4–24. CPRI Topologies



If the destination for the high-speed serial data that leaves the REC is the first RE, it is a single-hop connection. If the serial data from the REC must traverse through multiple REs before reaching the destination RE, it is a multi-hop connection.

Remotely locating the RF transceiver from the main base station introduces a complexity with overall system delay. The CPRI specification requires that the accuracy of measurement of roundtrip delay on single-hop and multi-hop connections be within ±16.276 ns to properly estimate the cable delay.

For a single-hop system, this allows a variation in roundtrip delay of up to ±16.276 ns. However, for multi-hop systems, the allowed delay variation is divided among the number of hops in the connection—typically, equal to ±16.276 ns/(the number of hops) but not always equally divided among the hops.

Deterministic latency on a CPRI link also enables highly accurate triangulation of the location of the caller.

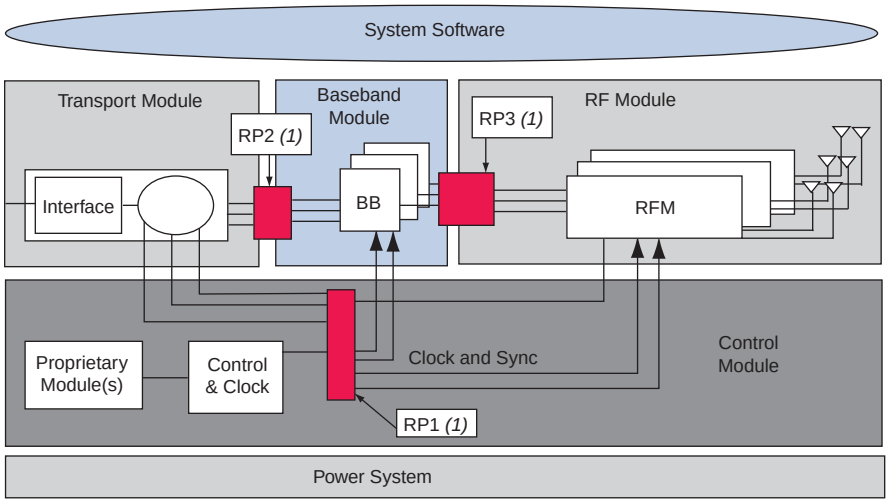
OBSAI was established by several OEMs to develop a set of specifications that can be used for configuring and connecting common modules into base transceiver stations (BTS).

The BTS has four main modules:

- Radio frequency (RF)
- Baseband
- Control
- Transport

Figure 4–25 shows a typical BTS. The radio frequency module (RFM) receives signals using portable devices and converts the signals to digital data. The baseband module processes the encoded signal and brings it back to the baseband before transmitting it to the terrestrial network using the transport module. A control module maintains the coordination between these three functions.

Figure 4–25. Example of the OBSAI BTS Architecture



Note to Figure 4–25:

(1) "RP" means Reference Point.

Using the deterministic latency option, you can implement the CPRI data rates in the following modes:

- 8/10-bit-width mode with one of the following:
  - 8-bit channel width
  - 10-bit channel width
- 16/20-bit-width mode with one of the following:
  - 16-bit channel width
  - 20-bit channel width

Table 4–8 lists sample channel width options, and CPRI and OBSAI data rates in the deterministic latency mode.

Table 4–8. Sample Channel Width Options for Supported Serial Data Rates

Serial Data Rate (Mbps)	Channel Width (FPGA-PCS Fabric)			
	8/10-Bit-Width		16/20-Bit-Width	
	8-Bit	16-Bit	16-Bit	32-Bit
614.4	Yes	Yes	—	—
1228.8	Yes	Yes	Yes	Yes
2457.6	—	Yes	Yes	Yes
3072	—	Yes	Yes	Yes
4915.2	—	—	—	Yes
6144	—	—	—	Yes
9830.4	—	—	—	Yes

For more information about deterministic latency PHY IP implementation, refer to the *Deterministic Latency PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

XAUI

This section describes XAUI link implementation using Stratix V transceivers. It provides the transceiver channel datapath description, clocking, and channel placement guidelines when configured in a XAUI configuration.

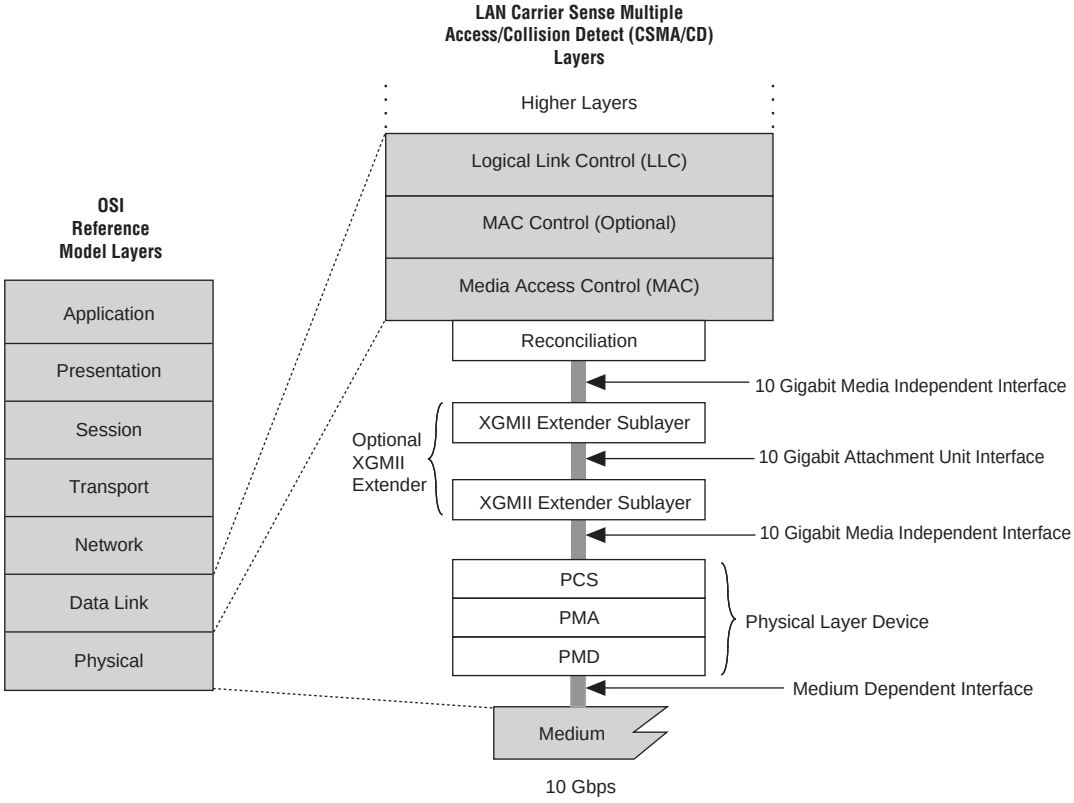
With the MegaWizard Plug-In Manager, you can implement a XAUI link. Under **Ethernet** in the Interfaces menu, select the **XAUI PHY** IP core. The XAUI PHY IP core implements the XAUI PCS in soft logic.

For more information, refer to the *XAUI PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

XAUI is a specific physical layer implementation of the 10 Gigabit Ethernet link defined in the IEEE 802.3ae-2002 specification. As shown in Figure 4–26, the XAUI PHY uses the XGMII interface to connect to the IEEE802.3 MAC and Reconciliation Sublayer (RS). The IEEE 802.3ae-2002 specification requires the XAUI PHY link to support a 10 Gbps data rate at the XGMII interface and four lanes each at 3.125 Gbps at the PMD interface.

Figure 4–26 shows the relationships between the XAUI PHY and other sublayers in the OSI reference model.

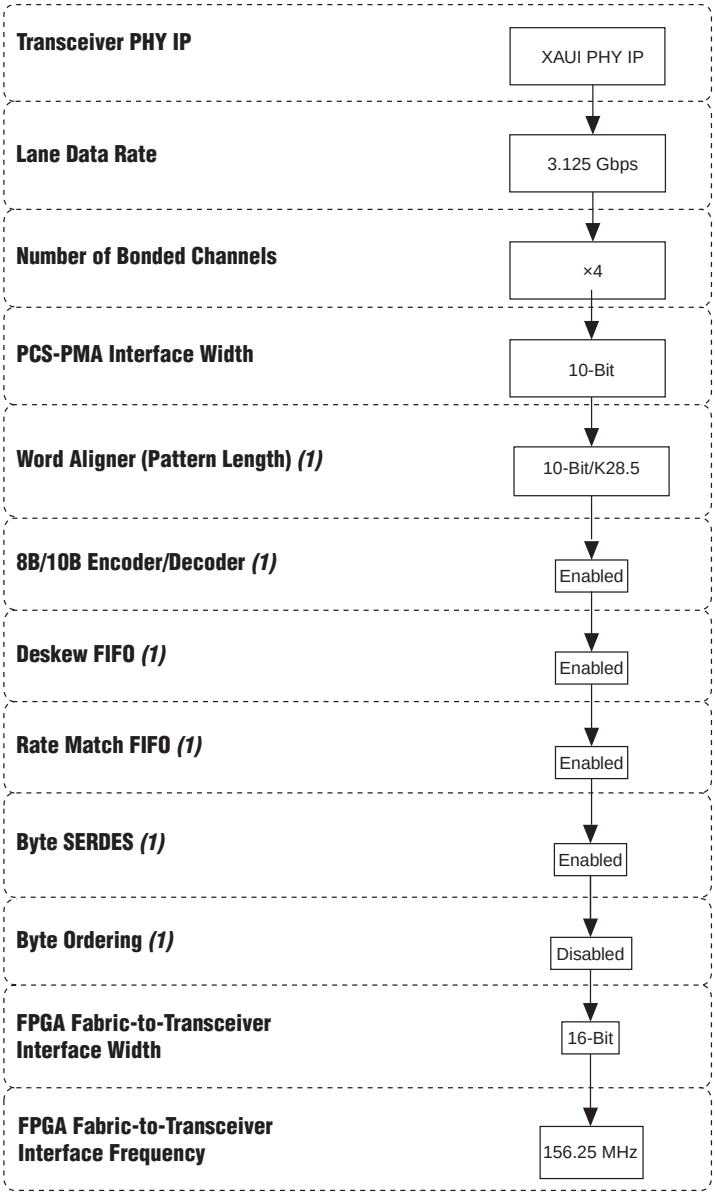
Figure 4–26. XAUI and XGMII Layers



Transceiver Datapath in a XAUI Configuration

Figure 4–27 shows the transceiver blocks enabled in a XAUI configuration. The XAUI PCS is implemented in soft logic inside the FPGA core when using the XAUI PHY IP core. You must ensure that your channel placement is compatible with the soft PCS implementation. For placement guidelines, refer to "Transceiver Channel Placement Guidelines" on page 4–52.

Figure 4–27. XAUI Datapath Configuration

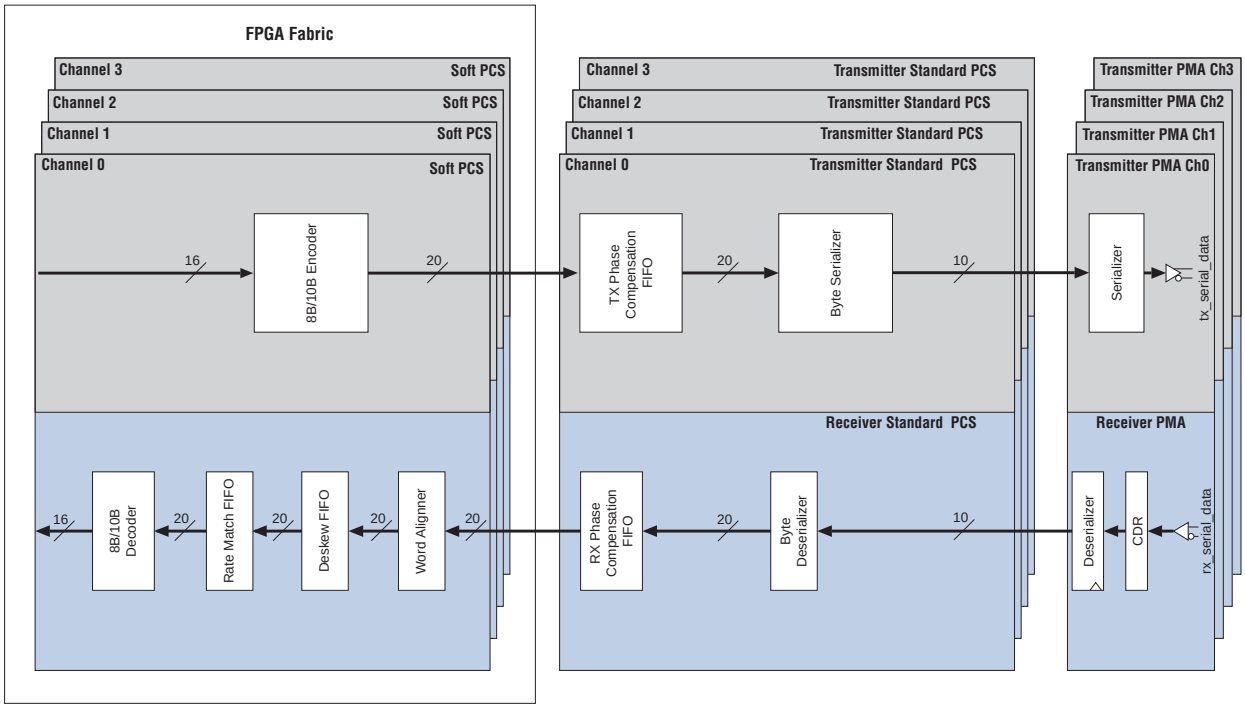


Note to Figure 4–27:

(1) Implemented in soft logic.

Figure 4–28 shows the transceiver datapath in a XAUI configuration.

Figure 4–28. Transceiver Channel Datapath for XAUI Configuration (1)



Note to Figure 4–28:

(1) Standard PCS in a low latency configuration is used in this configuration. Additionally, a portion of the PCS is implemented in soft logic.

Supported Features

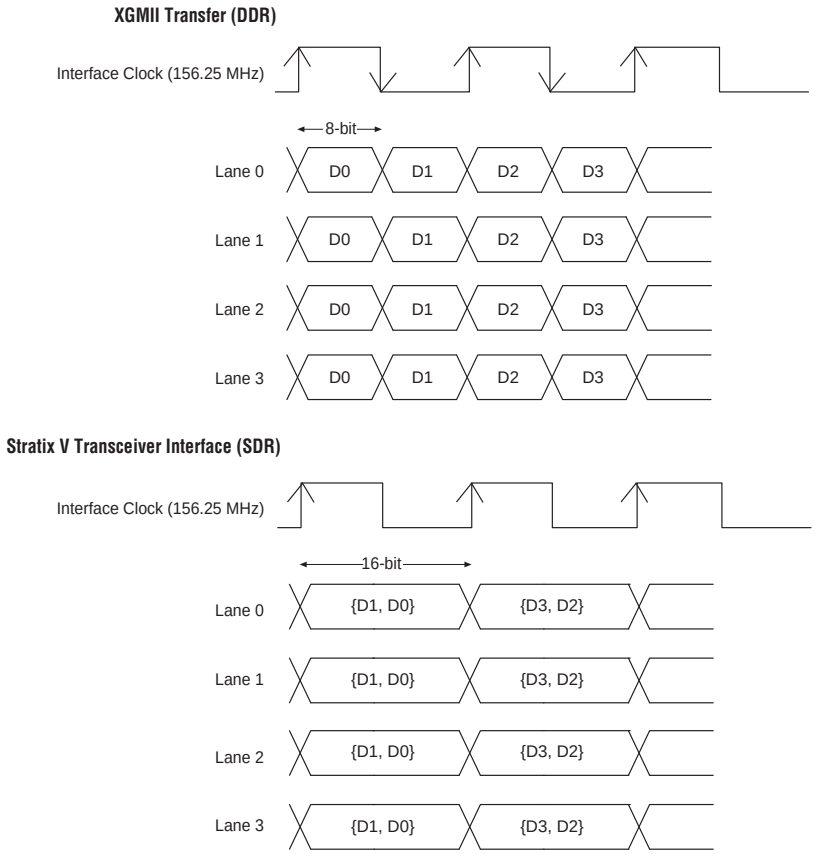
Stratix V transceivers support the following features in a XAUI configuration.

64-Bit SDR Interface to the MAC/RS

Clause 46 of the IEEE 802.3-2008 specification defines the XGMII interface between the XAUI PCS and the Ethernet MAC/RS. The specification requires each of the four XAUI lanes to transfer 8-bit data and 1-bit wide control code at both the positive and negative edge (DDR) of the 156.25 MHz interface clock.

Stratix V transceivers in a XAUI configuration do not support the XGMII interface to the MAC/RS as defined in IEEE 802.3-2008 specification. Instead, they allow the transferring of 16-bit data and 2-bit control code on each of the four XAUI lanes, only at the positive edge (SDR) of the 156.25 MHz interface clock, as shown in Figure 4–29.

Figure 4–29. Implementation of the XGMII Specification in Stratix V Devices



8B/10B Encoding/Decoding

Each of the four lanes in a XAUI configuration support an independent 8B/10B encoder/decoder as specified in Clause 48 of the IEEE802.3-2008 specification. 8B/10B encoding limits the maximum number of consecutive 1s and 0s in the serial data stream to five, thereby ensuring DC balance as well as enough transitions for the receiver CDR to maintain a lock to the incoming data.

The XAUI PHY IP core provides status signals to indicate running disparity as well as the 8B/10B code group error.

Transmitter and Receiver State Machines

In a XAUI configuration, the Stratix V transceivers implement the transmitter and receiver state diagrams shown in Figure 48-6 and Figure 48-9 of the IEEE802.3-2008 specification.

In addition to encoding the XGMII data to PCS code groups, in conformance with the 10GBASE-X PCS, the transmitter state diagram performs functions such as converting Idle ||I|| ordered sets into Sync ||K||, Align ||A||, and Skip ||R|| ordered sets.

In addition to decoding the PCS code groups to XGMII data, in conformance with the 10GBASE-X PCS, the receiver state diagram performs functions such as converting Sync ||K||, Align ||A||, and Skip ||R|| ordered sets to Idle ||I|| ordered sets.

Synchronization

The word aligner block in the receiver PCS of each of the four XAUI lanes implements the receiver synchronization state diagram shown in Figure 48-7 of the IEEE802.3-2008 specification.

The XAUI PHY IP core provides a status signal per lane to indicate if the word aligner is synchronized to a valid word boundary.

Deskew

The lane aligner block in the receiver PCS implements the receiver deskew state diagram shown in Figure 48-8 of the IEEE 802.3-2008 specification.

The lane aligner starts the deskew process only after the word aligner block in each of the four XAUI lanes indicates successful synchronization to a valid word boundary.

The XAUI PHY IP core provides a status signal to indicate successful lane deskew in the receiver PCS.

Clock Compensation

The rate match FIFO in the receiver PCS datapath compensates up to ±100 ppm difference between the remote transmitter and the local receiver. It does so by inserting and deleting Skip ||R|| columns, depending on the ppm difference.

The clock compensation operation begins after:

- The word aligner in all four XAUI lanes indicates successful synchronization to a valid word boundary.
- The lane aligner indicates a successful lane deskew.

The rate match FIFO provides status signals to indicate the insertion and deletion of the Skip ||R|| column for clock rate compensation.

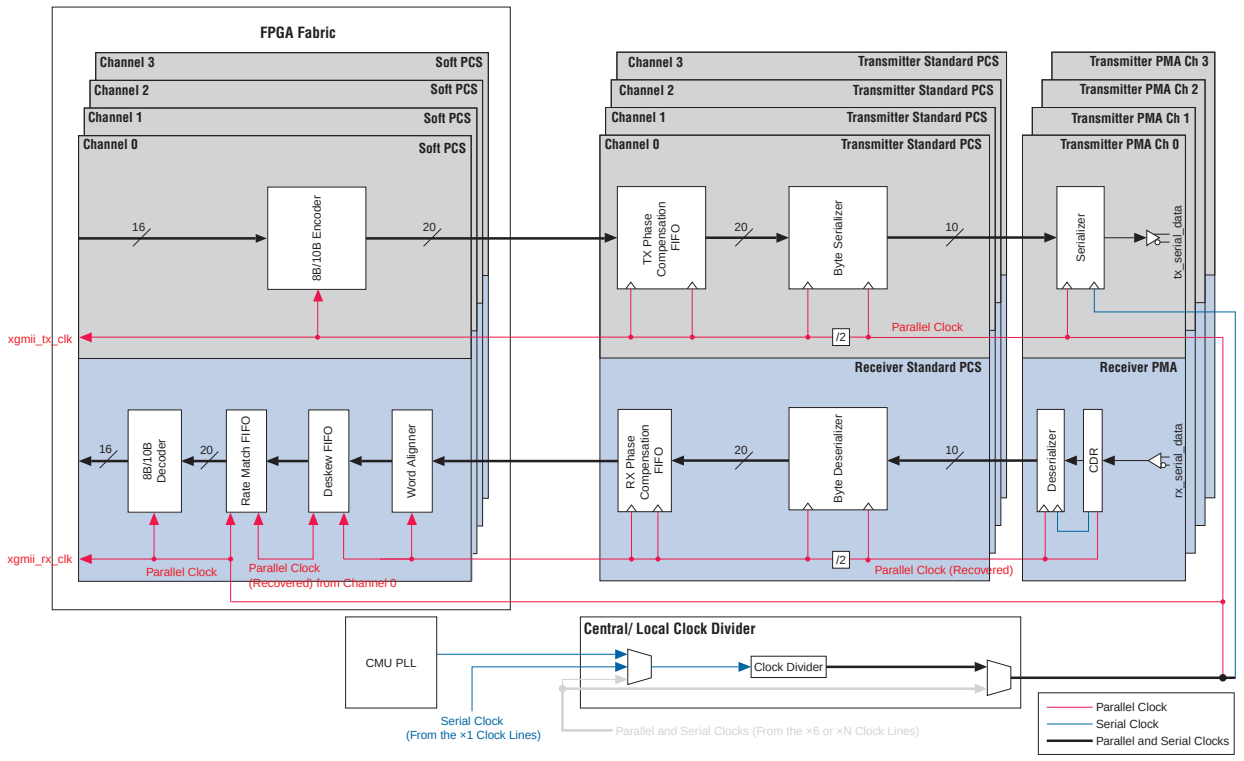
Transceiver Clocking and Channel Placement Guidelines

This section describes the transceiver clocking and channel placement guidelines for implementing the XAUI protocol in Stratix V devices.

Transceiver Clocking

Figure 4–30 shows the transceiver clocking in a XAUI configuration.

Figure 4–30. Transceiver Clocking Diagram for XAUI Configuration



In Figure 4–30, one of the two channel PLLs configured as a CMU PLL in a transceiver bank generates the transmitter serial and parallel clocks for the four XAUI channels. The x6 clock line carries the transmitter clocks to the PMA and PCS of each of the four channels.

Table 4–9 lists the input reference clock frequency, FPGA fabric-transceiver interface width, and interface frequency supported in a XAUI configuration.

Table 4–9. Input Reference Clock Frequency and Interface Speed Specifications for XAUI Configurations

Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Frequency (MHz)
156.25	16-bit data, 2-bit control	156.25

Transceiver Channel Placement Guidelines

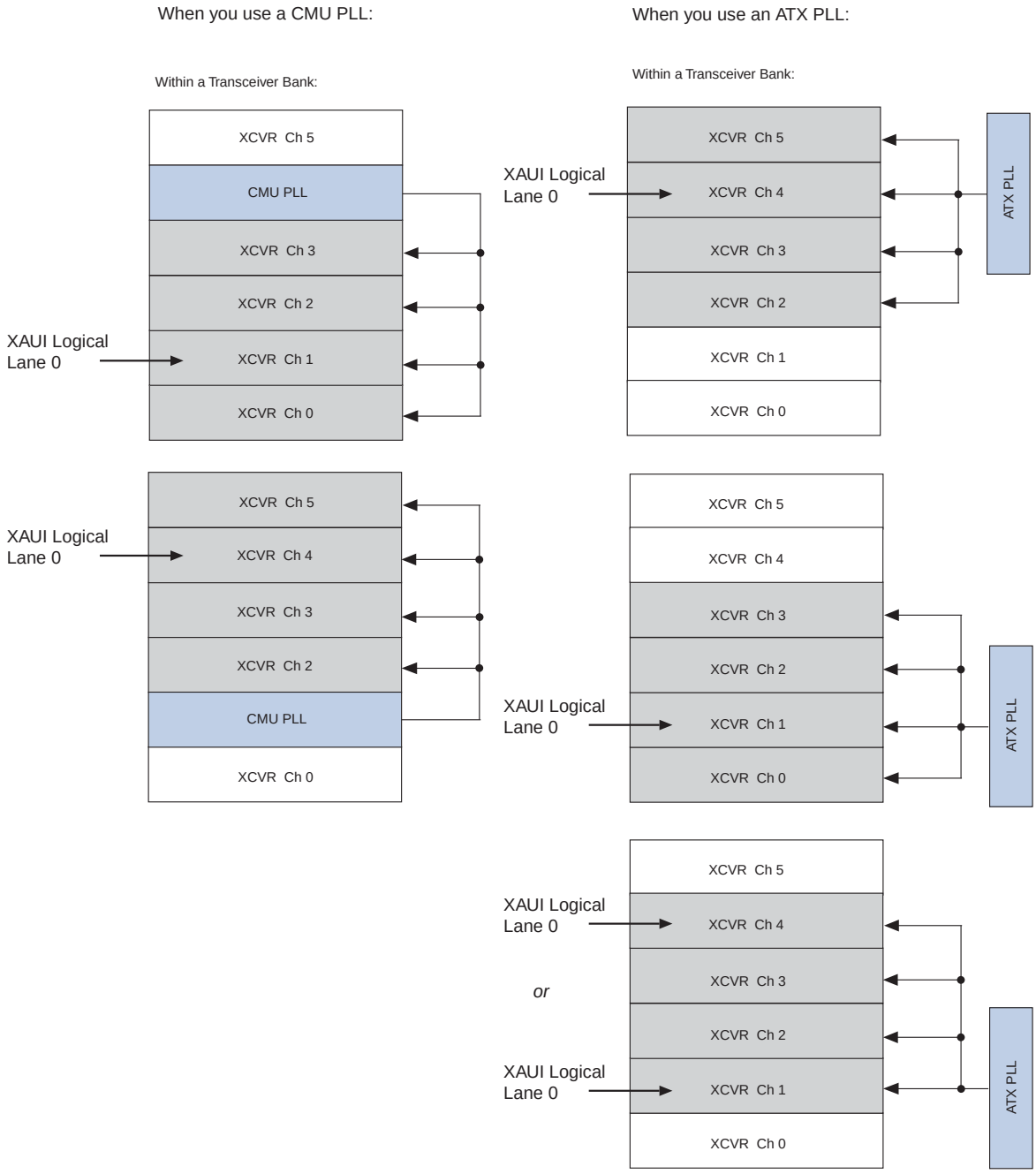
The soft PCS implementation of the XAUI configuration has placement restrictions. All four channels must be placed within one transceiver bank. A link cannot span two banks. You must place the channels according to the following guidelines.

Figure 4–31 shows the allowed channel placement when using either the CMU PLL or the ATX PLL to drive the XAUI link. The Quartus II software implements the XAUI PCS in soft logic and has Logical Lane 0 placement restrictions.

To implement the QSF assignment workaround using the Assignment Editor, refer to the *XAUI PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

For backward compatibility with earlier versions of the Quartus II software, ensure that your channels are placed in one of the configurations shown in Figure 4–31.

Figure 4–31. Transceiver Channel Placement Guidelines in a XAUI Configuration *(1)*



Note to Figure 4–31:

(1) A QSF assignment statement can be used to change the XAUI Logical Lane 0 assignment to a different transceiver channel.

Stratix V Device Configurations

Stratix V GX, GS, and GT devices offer the following configurations:

- "Standard PCS Configurations"
- "10G PCS Configurations"

These configurations allow you to modify, enable, or disable blocks based on your protocol requirements. This flexibility allows you to implement various protocols. Table 4–10 lists the protocols supported through the Custom and Low Latency PHY IPs.

Table 4–10. Custom and Low Latency PHY IPs

Protocol	PCS Type		IP
	Standard	10G	
ASI	Yes	—	Custom PHY and Native PHY
SATA/SAS	Yes	—	
GPON	Yes	—	
SFI 5.1	Yes	—	
SGMII	Yes	—	
GIGE	Yes	—	
SDI	Yes	—	
JESD204A	Yes	—	
OIF/CEI 11G-SR	Yes	Yes	Custom PHY, Low Latency PHY, and Native PHY
OIF/CEI 6G-SR	Yes	Yes	
100G Ethernet	—	Yes	Low Latency PHY and Native PHY
40G Ethernet	—	Yes	
SFI-S	—	Yes	
10GPON/EPON	—	Yes	
SDI 10G	—	Yes	

Standard PCS Configurations

You can configure the standard PCS to have one of the following datapaths:

- "Custom Datapath"—you define your custom datapath configurations by selecting the blocks to use and the appropriate data width. You must use the Custom PHY IP to enable the standard PCS in custom datapath. In the MegaWizard Plug-In Manager, you can implement a Custom PHY link by instantiating the **Custom PHY IP** under **Transceiver PHY** in the Interfaces menu.

For details about implementing a custom datapath with the Custom PHY IP, refer to the *Custom PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

- "Low Latency Datapath"—a low latency datapath bypasses much of the standard PCS, allowing more design control in the FPGA fabric. You must use the Low Latency PHY IP to enable the standard PCS in a low latency datapath. In the MegaWizard Plug-In Manager, you can implement a Low Latency PHY link by instantiating the **Low Latency PHY IP** under **Transceiver PHY** in the Interfaces menu. In the Low Latency GUI under the **General** tab, select **Standard** on the Datapath type selection.

  For details about implementing a low latency datapath with the Low Latency PHY IP, refer to the *Low Latency PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

The following sections describe these datapaths in more detail.

Custom Datapath

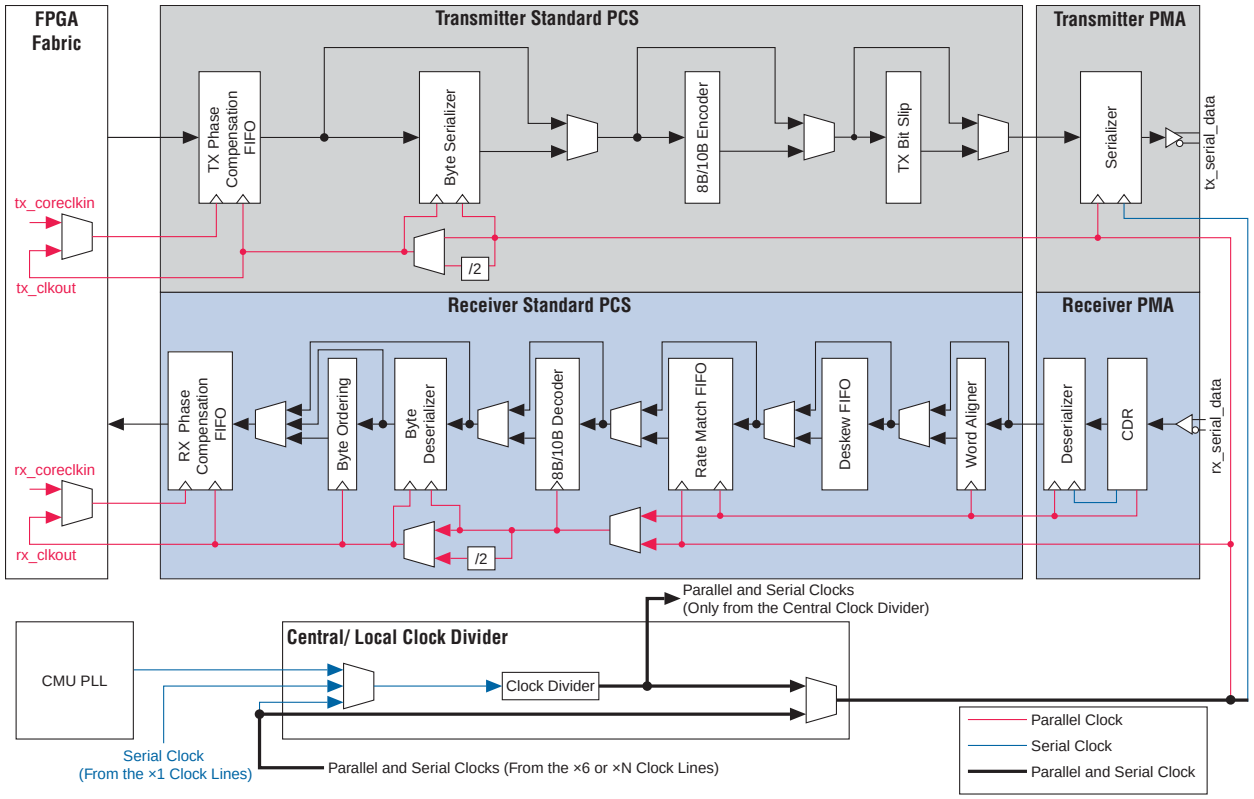
The custom datapath consists of the following blocks:

- 8B/10B encoder and decoder
- Word aligner
- Deskew FIFO
- Rate match FIFO (clock rate compensation FIFO)
- Byte ordering block
- Phase compensation FIFO
- Byte serializer and deserializer
- Transmit bit slip

Refer to the "Standard PCS Architecture" section in the *Transceiver Architecture in Stratix V Devices* chapter for the architectural details of these blocks.

Figure 4–32 shows the complete datapath and clocking for the custom transceiver configuration with the standard PCS.

Figure 4–32. Standard PCS Custom Datapath



You can divide the custom datapath into two configurations based on the FPGA fabric-transceiver interface width and the PMA-PCS interface width (serialization factor):

- **Custom 8/10-bit-width**—the PCS-PMA interface width is in 8-bit or 10-bit mode for lower data rates.
- **Custom 16/20-bit-width**—the PCS-PMA interface width is in 16-bit or 20-bit mode for higher data rates.

Table 4–11 lists the supported data rates.

Table 4–11. PCS-PMA Interface Widths and Data Rates in Custom 8/10-Bit-Width and 16/20-Bit-Width Modes for Stratix V Devices

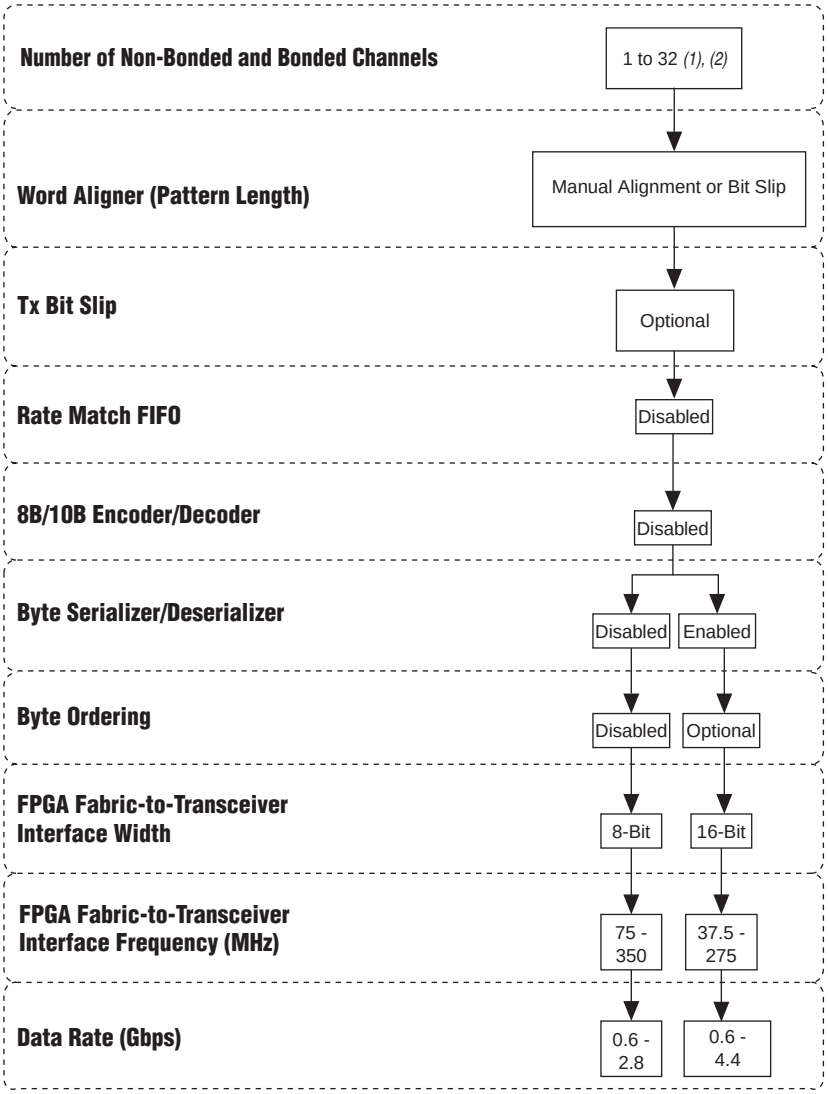
PCS-PMA Interface Width	Supported Data Rate Range PMA
Custom 8-bit width	622 Mbps to 4.4 Gbps
Custom 10-bit width	622 Mbps to 5.5 Gbps
Custom 16-bit width	1 Gbps to 8.8 Gbps
Custom 20-bit width	1 Gbps to 11 Gbps

The custom datapath options available at different speeds are shown in Figure 4–33 through Figure 4–36.

Figure 4–33 shows the available options for the standard PCS custom 8-bit PMA-PCS interface width. The maximum frequencies shown in Figure 4–33 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–33. Standard PCS Custom 8-Bit PMA-PCS Interface Width



Notes to Figure 4–33:

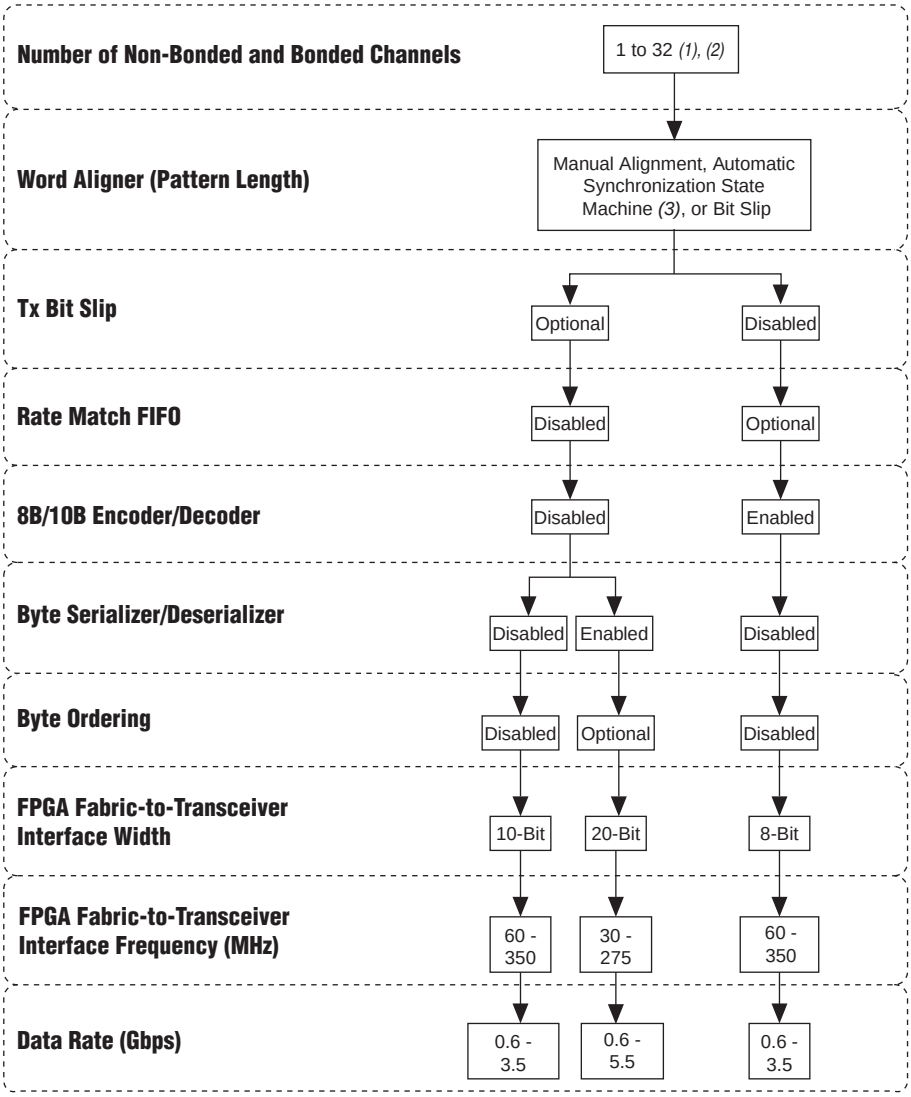
(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Figure 4–34 shows the available options for the standard PCS custom 10-bit PMA-PCS interface width. The maximum frequencies shown in Figure 4–34 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–34. Standard PCS Custom 10-Bit PMA-PCS Interface Width



Notes to Figure 4–34:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

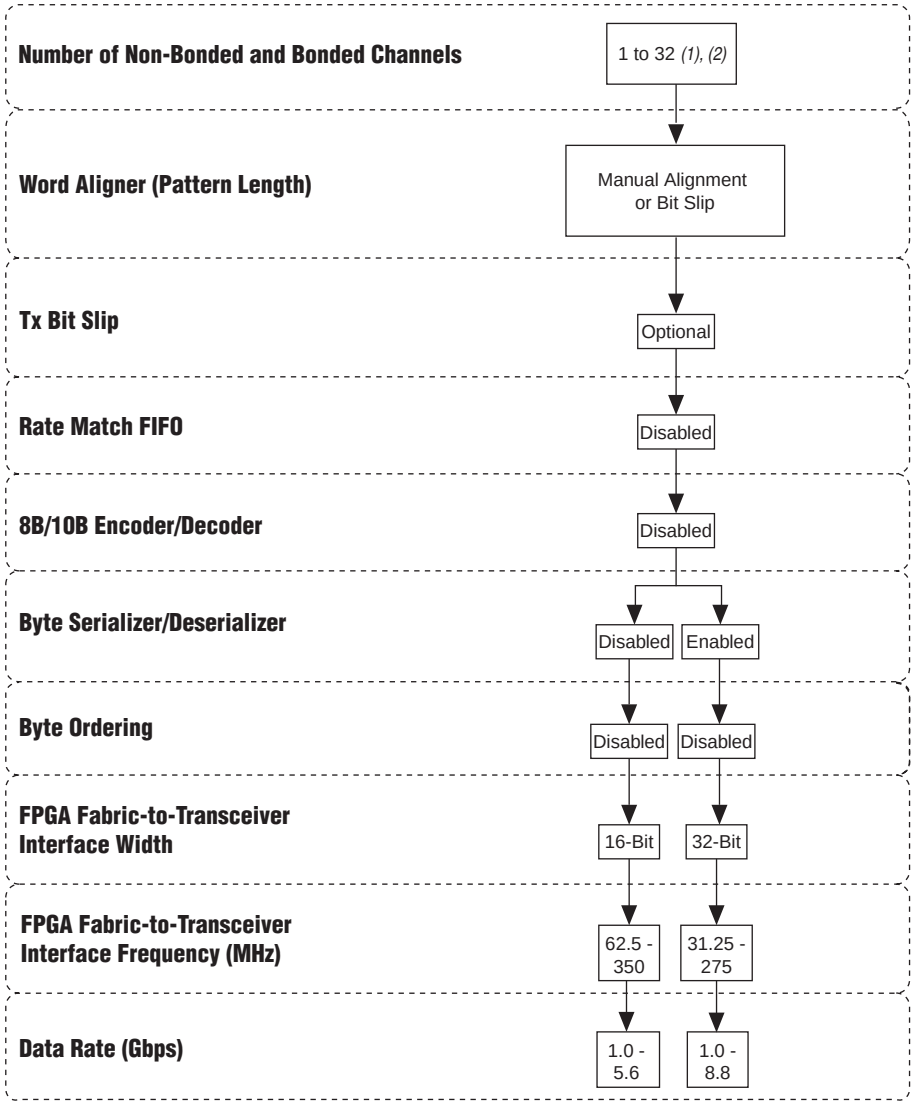
(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) Automatic Synchronization State Machine requires enabling the 8B/10B Encoder/Decoder.

Figure 4–35 shows the available options for the standard PCS custom 16-bit PMA-PCS interface width. The maximum frequencies shown in Figure 4–35 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–35. Standard PCS Custom 16-Bit PMA-PCS Interface Width



Notes to Figure 4–35:

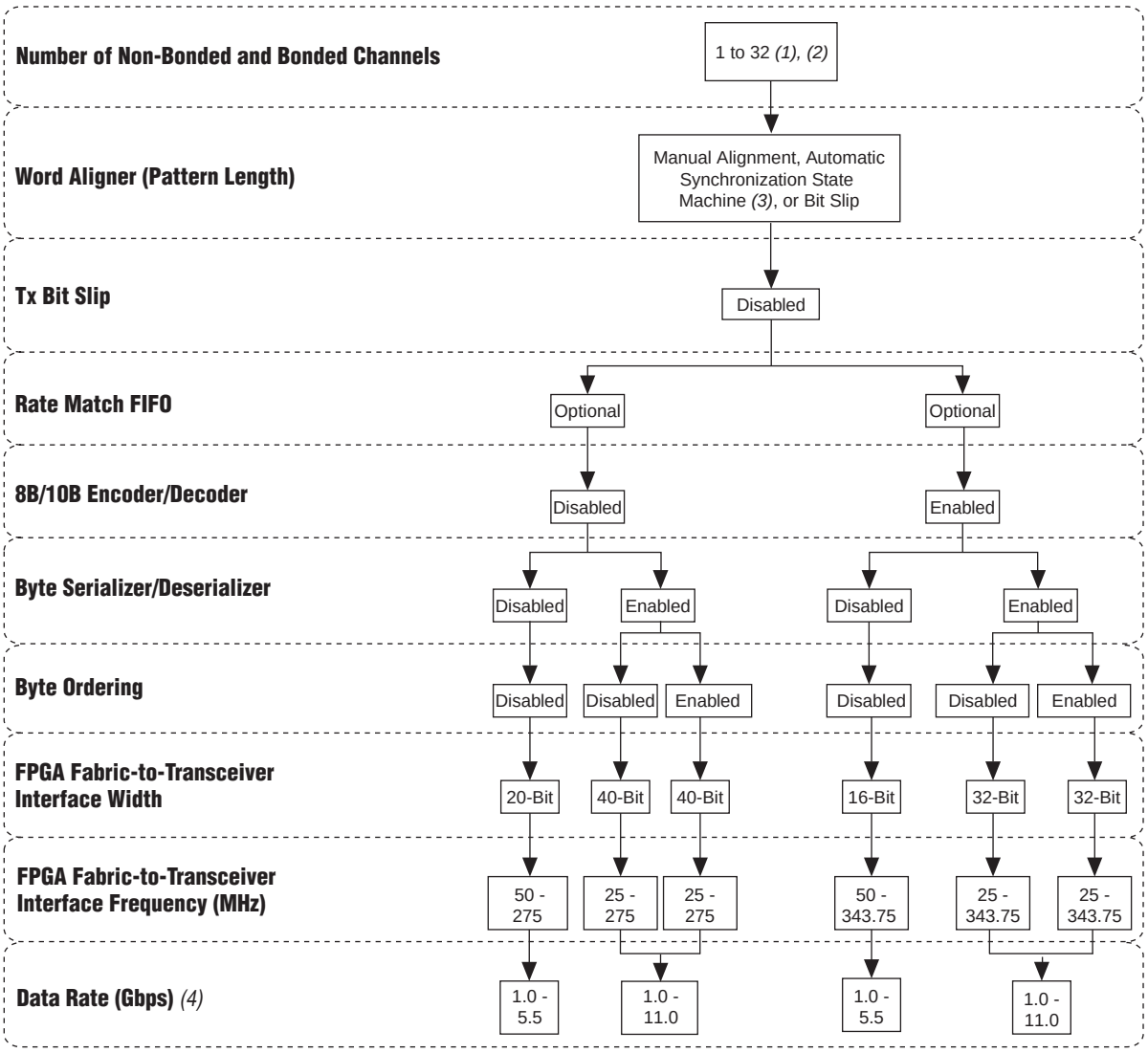
(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Figure 4–36 shows the available options for the standard PCS custom 20-bit PMA-PCS interface width. The maximum frequencies shown in Figure 4–36 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–36. Standard PCS Custom 20-Bit PMA-PCS Interface Width



Notes to Figure 4–36:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) Automatic Synchronization State Machine requires enabling the 8B/10B Encoder/Decoder.

(4) The maximum data rate specification shown in Figure 4–36 is valid only for the -2 (fastest) speed grade devices. For data rate specifications for other speed grades, refer to the *Stratix V Device Datasheet*.

Low Latency Datapath

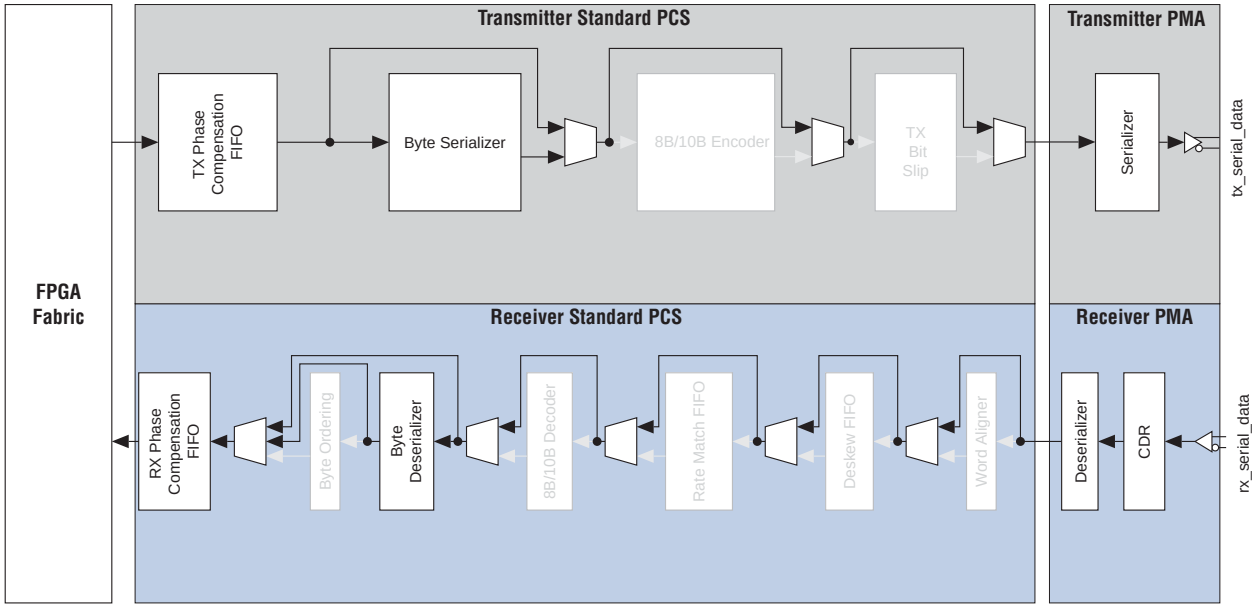
The standard PCS can be used in a low latency datapath that contains only the following blocks:

- Phase compensation FIFO
- Byte serializer and deserializer

Refer to the "Standard PCS Architecture" section in the *Transceiver Architecture in Stratix V Devices* chapter for the architectural details of these blocks.

Figure 4–37 shows the complete datapath for the low latency transceiver configuration with the standard PCS.

Figure 4–37. Standard PCS Low Latency Datapath



You can divide the low latency datapath into two configurations based on the FPGA fabric-transceiver interface width and the PMA-PCS interface width (serialization factor):

- **Low latency 8/10-bit-width**—the PCS-PMA interface width is in 8-bit or 10-bit mode for lower data rates.
- **Low latency 16/20-bit-width**—the PCS-PMA interface width is in 16-bit or 20-bit mode for higher data rates.

Table 4–12 lists the PCS and PMA interface widths and data rates in low latency single-width and double-width modes.

Table 4–12. PCS-PMA Interface Widths and Data Rates (Part 1 of 2)

Low Latency PHY IP Core	Supported Data Rate Range PMA
Low latency 8-bit width	600 Mbps to 4.4 Gbps
Low latency 10-bit width	600 Mbps to 5.5 Gbps

Table 4–12. PCS-PMA Interface Widths and Data Rates (Part 2 of 2)

Low Latency PHY IP Core	Supported Data Rate Range PMA
Low latency 16-bit width	1 Gbps to 8.8 Gbps
Low latency 20-bit width	1 Gbps to 11.0 Gbps

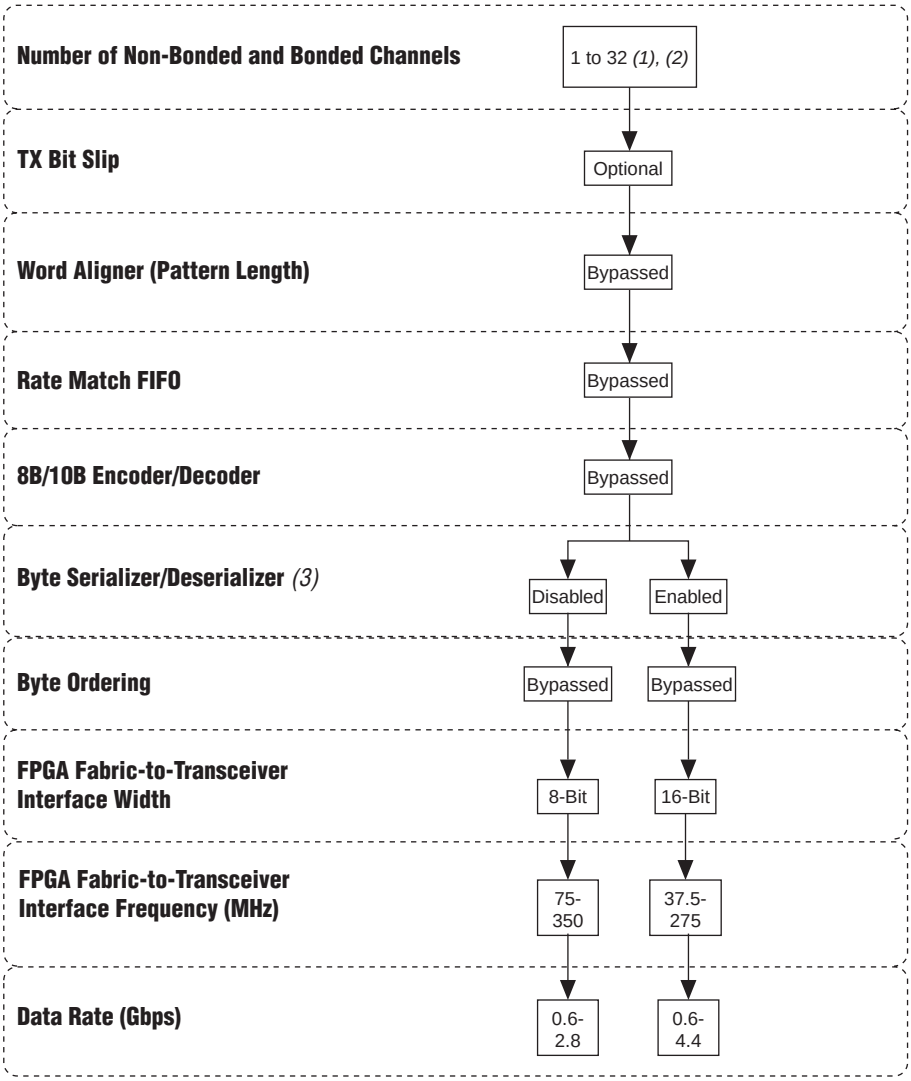
In the low latency datapath, the TX and RX phase compensation FIFOs are always enabled. Depending on the targeted data rate, you may bypass the byte serializer and deserializer blocks.

The low latency datapath options available at different speeds are shown in Figure 4–38 to Figure 4–41.

Figure 4–38 shows the available options for the standard PCS low latency 8-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies shown in Figure 4–38 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–38. Standard PCS Low Latency 8-Bit PMA-PCS Interface Width



Notes to Figure 4–38:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

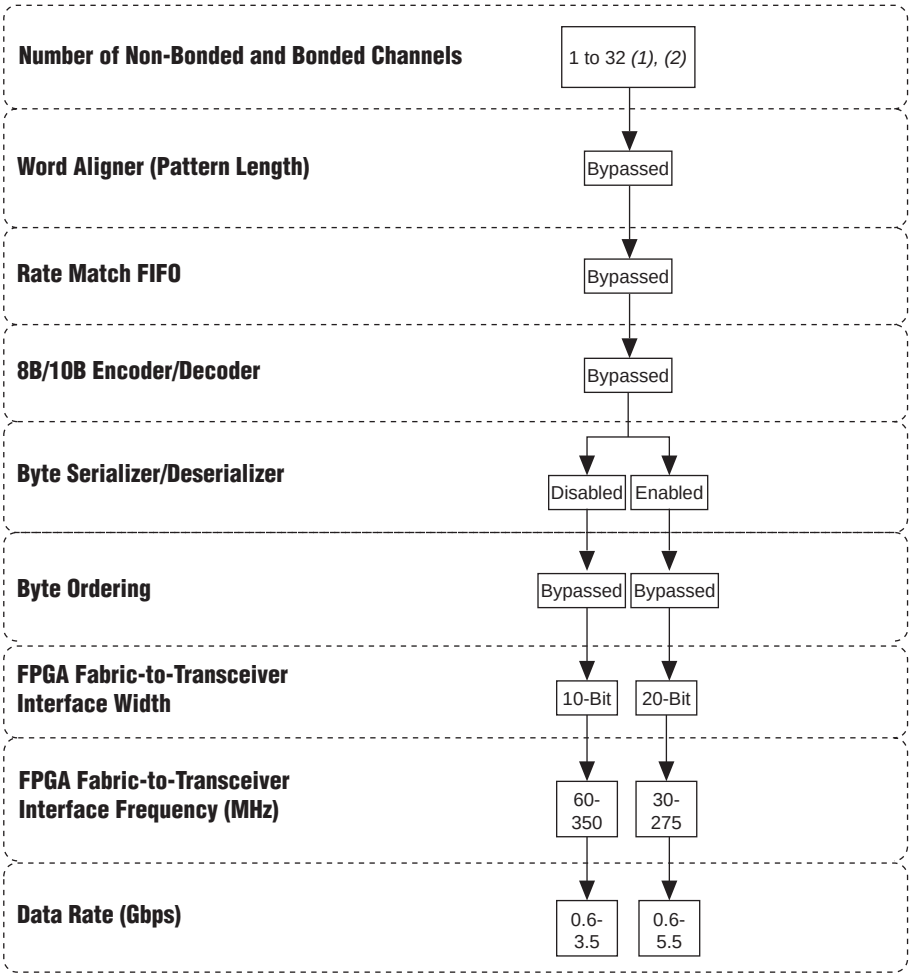
(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(3) The Quartus II software selects whether the byte serializer/deserializer is enabled or disabled based on the datapath width.

Figure 4–39 shows the available options for the standard PCS low latency 10-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies shown in Figure 4–39 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–39. Standard PCS Low Latency 10-Bit PMA-PCS Interface Width



Notes to Figure 4–39:

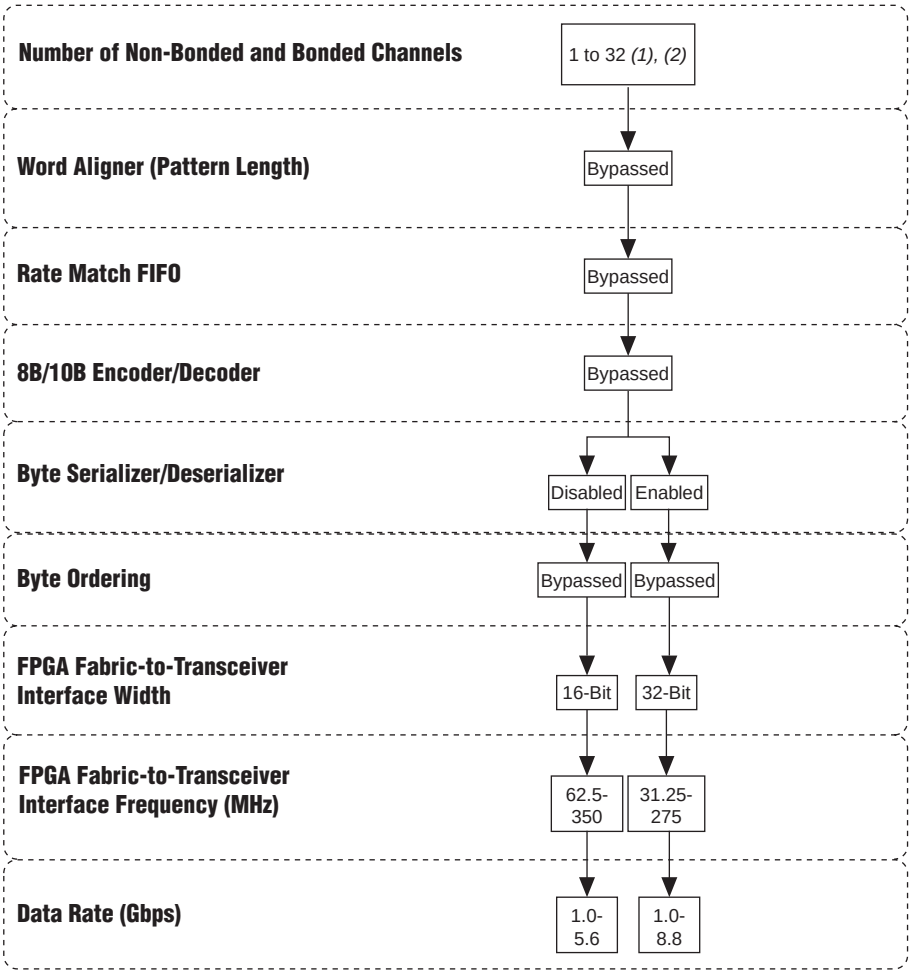
(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Figure 4–40 shows the available options for the standard PCS low latency 16-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies shown in Figure 4–40 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–40. Standard PCS Low Latency 16-Bit PMA-PCS Interface Width



Notes to Figure 4–40:

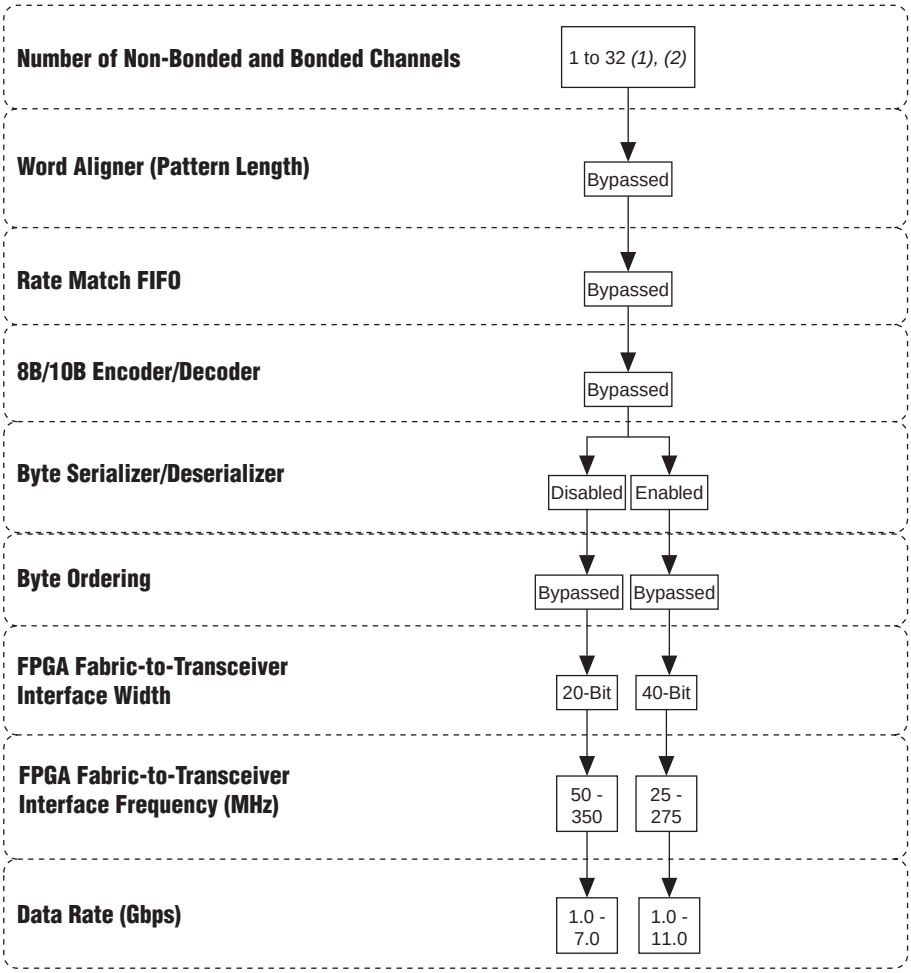
(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

Figure 4–41 shows the available options for the standard PCS low latency 20-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies shown in Figure 4–41 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *Stratix V Device Datasheet*.

Figure 4–41. Standard PCS Low Latency 20-Bit PMA-PCS Interface Width



Notes to Figure 4–41:

(1) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(2) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

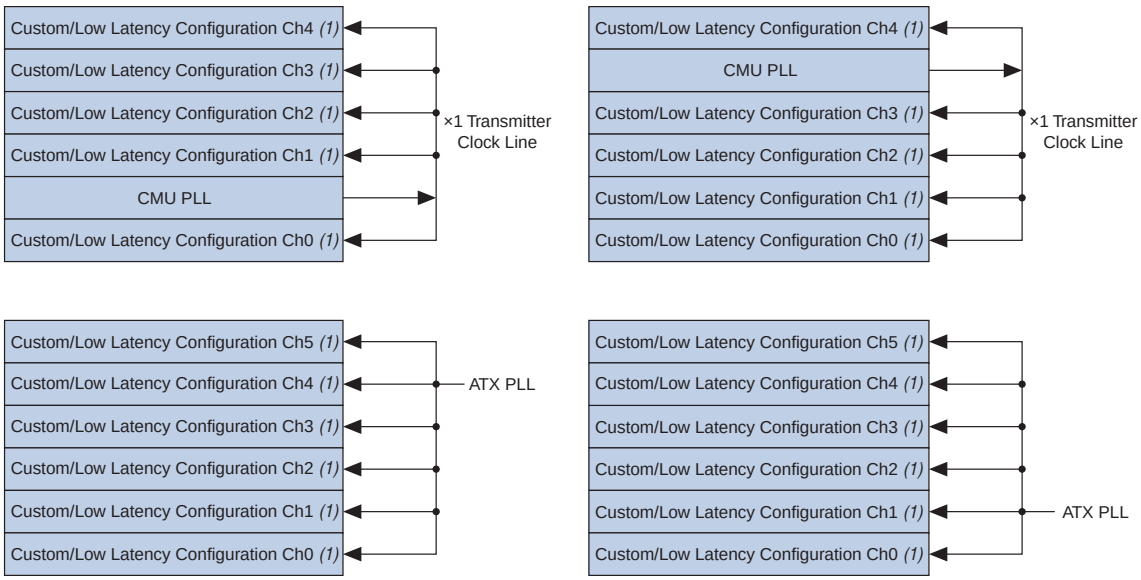
Transceiver Channel Placement Guidelines

Stratix V devices allow the placement of up to five channels when a CMU PLL is used or up to six channels when an ATX PLL is used in a non-bonded configuration within the same transceiver bank:

- Custom PHY IP with standard PCS datapath configuration
- Low Latency PHY IP with Standard PCS or 10G PCS (same data rate) in low latency datapath configuration

Figure 4–42 shows a non-bonded supported channel placement scenario using CMU and ATX PLLs.

Figure 4–42. Non-Bonded Channel Placement Guidelines with Standard and 10G PCS in Custom and Low Latency Datapath Configurations *(1)*



Note to Figure 4–42:

(1) All channels shown in Figure 4–42 are assumed to contain a transmitter and receiver.

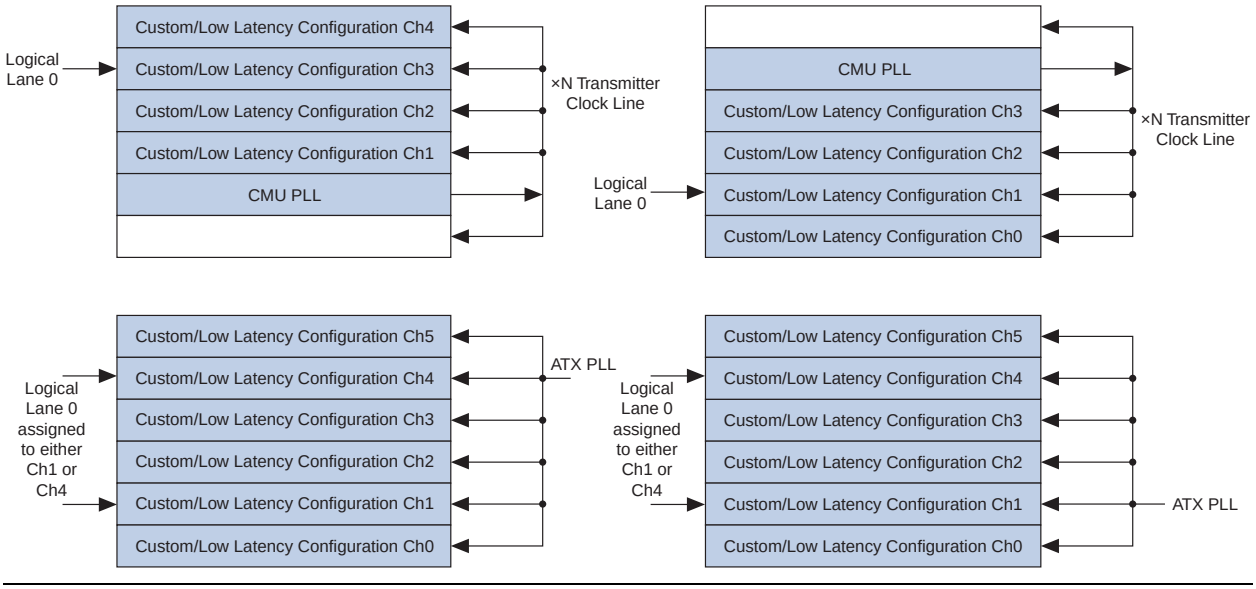
Figure 4–43 shows a xN bonded supported channel placement scenario using CMU and ATX PLLs within a transceiver bank.

Stratix V devices allow the placement of up to four channels when a CMU PLL is used or up to six channels when an ATX PLL is used in a bonded configuration within the same transceiver bank:

- Custom PHY IP with standard PCS datapath configuration
- Low Latency PHY IP with Standard PCS or 10G PCS (same data rate) in low latency datapath configuration

The xN bonding method requires Logical Lane 0 be placed at either transceiver physical channel 1 or 4 within a transceiver bank. The PLL feedback compensation bonding method does not have a Logical Lane 0 assignment requirement and must be used when more than one transceiver bank is needed. However, PLL feedback compensation bonding requires the use of one PLL per transceiver bank.

Figure 4–43. Bonded Channel Placement Guidelines with Standard and 10G PCS in Custom and Low Latency Datapath Configurations


Logical Lane 0
Custom/Low Latency Configuration Ch4
Custom/Low Latency Configuration Ch3
Custom/Low Latency Configuration Ch2
Custom/Low Latency Configuration Ch1
CMU PLL
×N Transmitter Clock Line
CMU PLL
Custom/Low Latency Configuration Ch3
Custom/Low Latency Configuration Ch2
Logical Lane 0
Custom/Low Latency Configuration Ch1
Custom/Low Latency Configuration Ch0
×N Transmitter Clock Line
Logical Lane 0 assigned to either Ch1 or Ch4
Custom/Low Latency Configuration Ch5
Custom/Low Latency Configuration Ch4
Custom/Low Latency Configuration Ch3
Custom/Low Latency Configuration Ch2
Custom/Low Latency Configuration Ch1
Custom/Low Latency Configuration Ch0
ATX PLL
Logical Lane 0 assigned to either Ch1 or Ch4
Custom/Low Latency Configuration Ch5
Custom/Low Latency Configuration Ch4
Custom/Low Latency Configuration Ch3
Custom/Low Latency Configuration Ch2
Custom/Low Latency Configuration Ch1
Custom/Low Latency Configuration Ch0
ATX PLL

10G PCS Configurations

You can only configure 10G PCS in the low latency datapath.

In the MegaWizard Plug-In Manager, you can implement a Low Latency PHY link with the 10G PCS by instantiating the **Low Latency PHY IP** under **Transceiver PHY** in the Interfaces menu. In the Low Latency GUI under the **General** tab, select **10G** on the **Datapath type** field.

A Low Latency PHY IP core with the 10G PCS is available for 32-bit, 40-bit, 50-bit, 64-bit, or 66-bit PCS data width configurations. Figure 4–44 shows the 10G Low Latency configuration datapath.

Figure 4–44. 10G PCS Low Latency Configuration Datapath

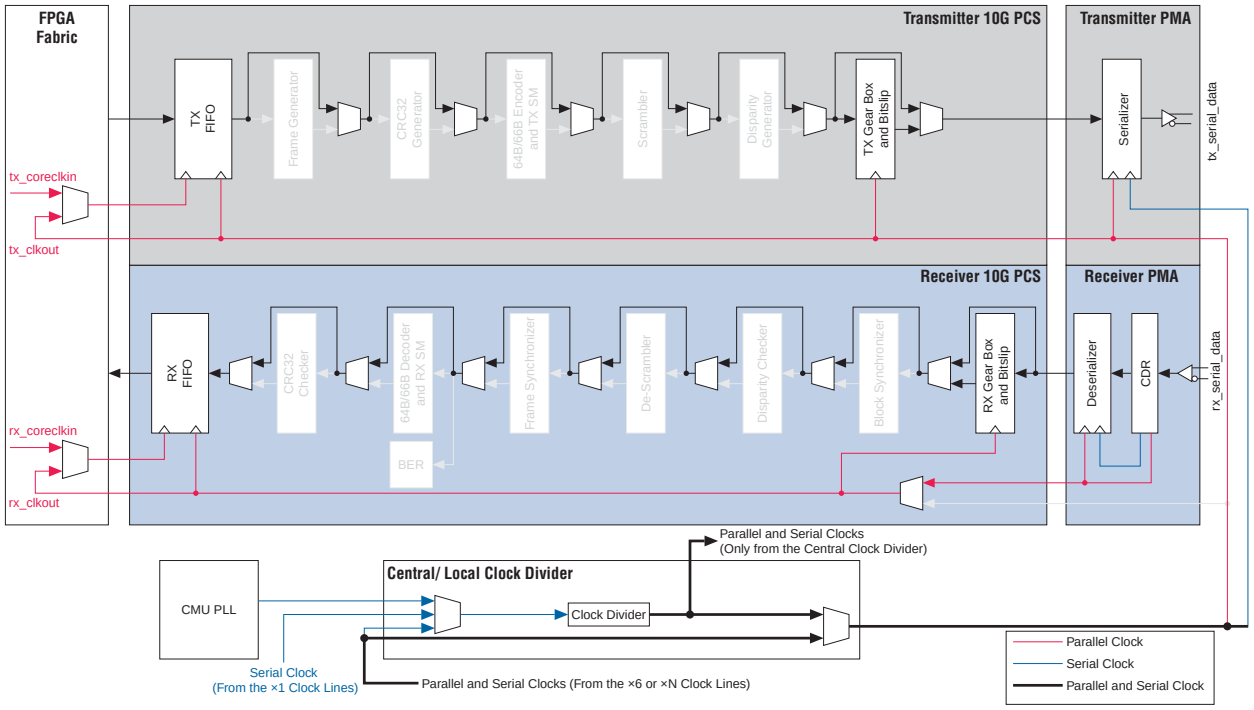


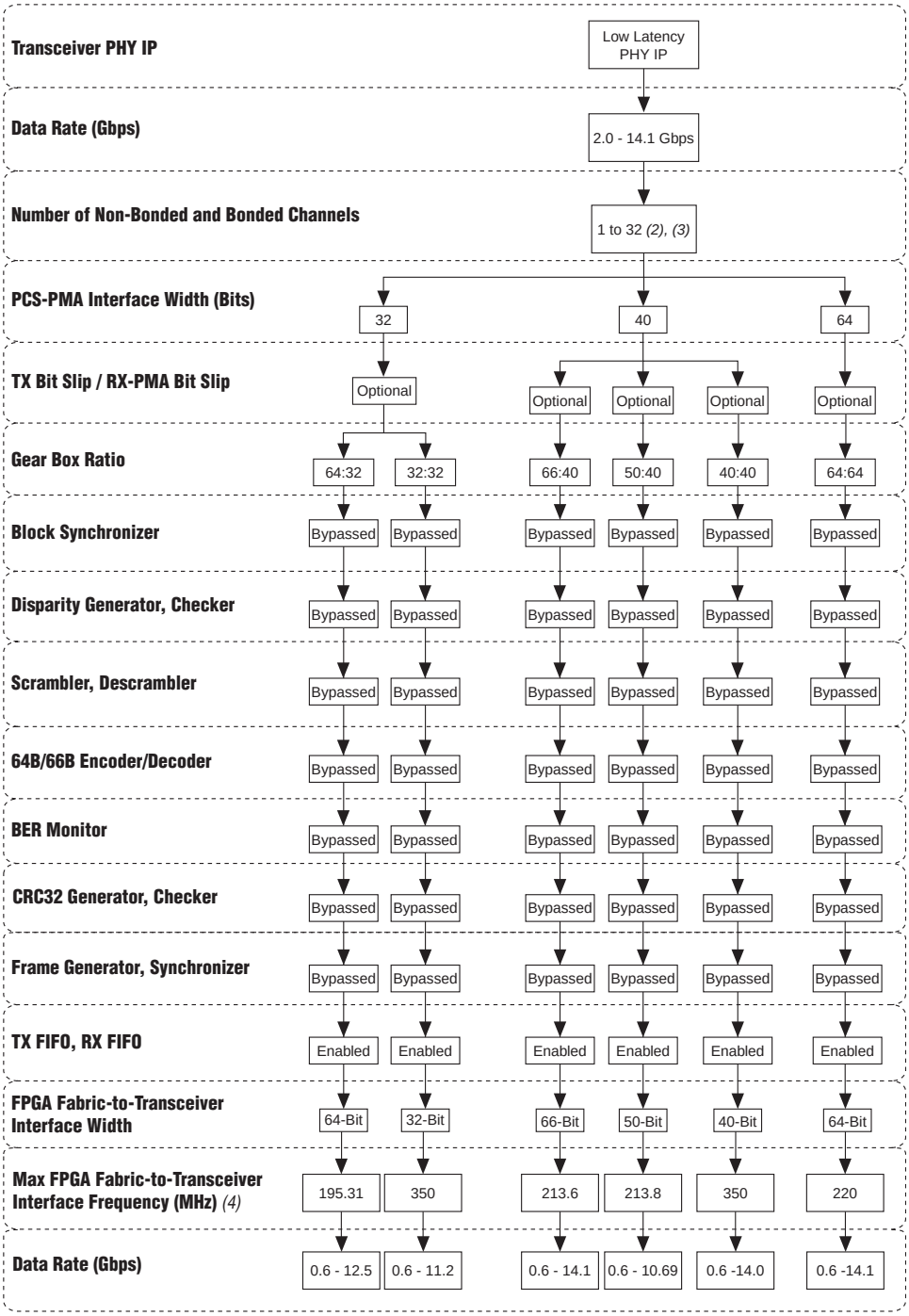
Figure 4–45 shows the different options available when you configure the 10G PCS in the low latency configuration.

For more information about the Low Latency PHY IP core, refer to the *Low Latency PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

The blocks shown as "Disabled" are not used but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The FPGA fabric-to-transceiver interface frequency specified in Figure 4–45 is for maximum speed grade devices.

For the limits of all speed grades, refer to the *Stratix V Device Datasheet*.

Figure 4–45. Options for 10G PCS Low Latency Configuration *(1)*



Notes to Figure 4–45:

(1) For the limits of all speed grades, refer to the *"Transceiver Specifications for Stratix V GX and GS Devices"* table in the *Stratix V Device Datasheet*.

(2) For xN bonding, the number of bonded channels is up to four using CMU PLL and up to six using ATX PLL, provided the data rate is supported by the CMU PLL and ATX PLL.

(3) Bonding more than six channels requires PLL feedback compensation bonding. PLL feedback compensation bonding requires one PLL per transceiver bank and the PLL reference clock frequency must have the same value as the lane data rate divided by the serialization factor.

(4) You must generate an `rx_coreclkin` with the specified frequency whenever the gear box is enabled.

The Quartus II software supports both non-bonded configuration and bonded configurations up to 32 lanes in the link when the 10G PCS in low latency datapath configuration is enabled. If you create multiple non-bonded channels with the 10G PCS in low latency mode, a common parallel clock (used in the bonded lane or channel configuration) is not generated by the central clock divider block. Each transmitter channel takes the high-speed clock, generated by the channel PLL, and locally divides it to generate the parallel clock.

For more information, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

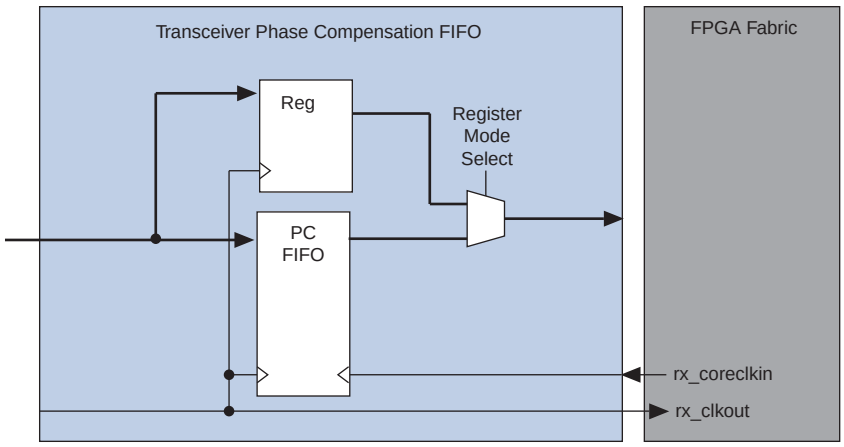
Datapath Functionality

This section discusses the different 10G PCS blocks used when you implement the 10G PCS in low latency mode.

Transmitter and Receiver FIFO

The FIFOs can be configured in phase compensation or registered mode, as shown in Figure 4–46 for the RX path. In phase compensation mode, the FIFO compensates the phase differences in the clock between the read and write side of the FIFO. The clocking scheme for the write side of the transmitter (TX) and receiver (RX) FIFOs depends on whether the gear box is enabled and on its ratio (40:66, 40:50, or 32:64). The clocking scheme is described in the "Clocking" section.

Figure 4–46. Phase Compensation FIFO in RX Path



Gear Box

The gear box translates the datapath width differences between the PCS and the physical medium attachment (PMA) interfaces. The gear box contains handshake control logic and FIFOs to implement the data-width translation. For the supported gear box ratio, refer to Figure 4–45.

TX Bit Slip Feature

The bit slip feature allows you to slip the transmitter side bits before they are sent to the gear box. The number of bits slipped is equal to the FPGA fabric-to-transceiver interface width minus 1. For example, if the FPGA fabric-to-transceiver interface width is 64 bits, a maximum of 63 bits can be slipped. That is, `bit[63]` from the first word and `bit[62:0]` are concatenated to form a 64 bit word (`bit[62:0]` from the second word, `bit[63]` from the first word LSB). The 7-bit input control signal is available to the FPGA fabric. For a 63-bit shift mentioned above, set the value of the input control to 7'b0011111.

Clocking

This section describes the transceiver datapath clocking. Figure 4–47 shows the clocking scheme when the gear box ratio is 64:64, 40:40, or 32:32. There is no frequency difference between the read and write side of the TX and RX FIFO clocks because the gear box is the same ratio. The Quartus II software automatically connects the clocks to the read and write side of the TX FIFO and RX FIFO. In this configuration, the data from the TX FIFO is still fed to the gear box before being sent to the serializer. The gear box cannot be bypassed or disabled.

Figure 4–47. 10G PCS Low Latency Datapath with Gear Box in 64:64, 40:40, and 32:32 Ratio

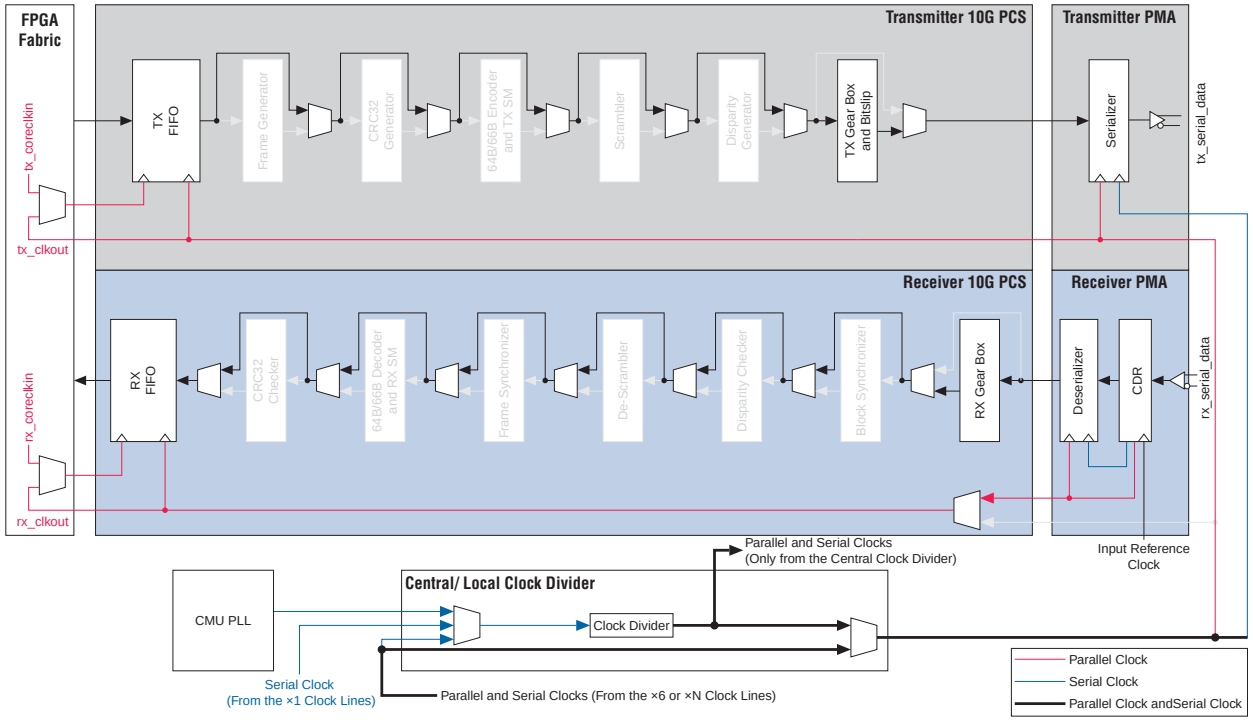


Figure 4–48 shows the configuration where the gear box ratio is 64:32. The FPGA fabric interface width (64 bits) is exactly twice the internal transceiver datapath width. You can divide the tx_clkout and rx_clkout in the FPGA fabric by two, and use them to clock the write side of TX FIFO and the read side of RX FIFO, respectively. Select the tx_coreclkin and the rx_coreclkin ports in the Low Latency PHY IP core and connect the divided clock to these ports, as shown in Figure 4–48.

Figure 4–48. 10G PCS Low Latency Datapath with the Gear Box Ratio of 64:32

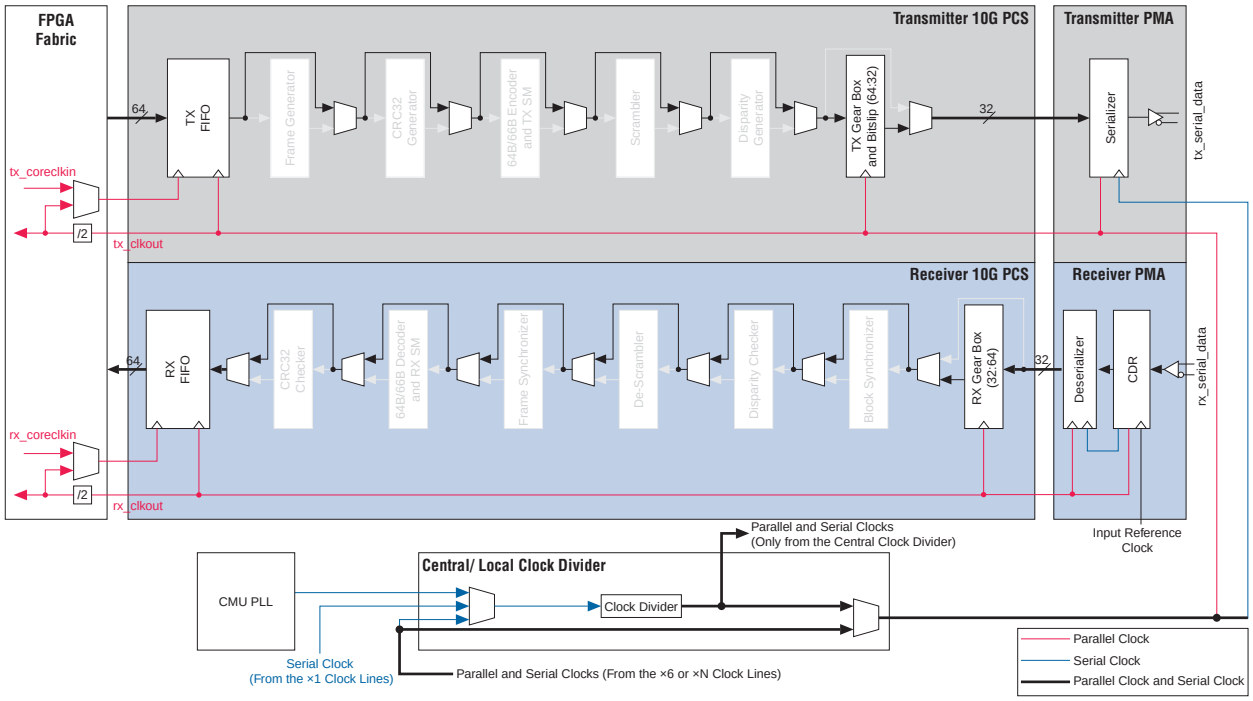


Figure 4–49 shows the clocking scheme when the gear box ratio is 66:40. For a gear box ratio of 66:40, the `rx_clkout` parallel clock provided is a recovered clock coming from the CDR with a divided-by-66 output frequency.

The `tx_clkout` parallel clock is generated from the transmit PLL feeding a fractional PLL ("fPLL" in Figure 4–49) that is automatically instantiated from the FPGA core with a divided-by-66 output frequency.

Figure 4–49. 10G PCS Low Latency Datapath with the Gear Box Ratio of 66:40

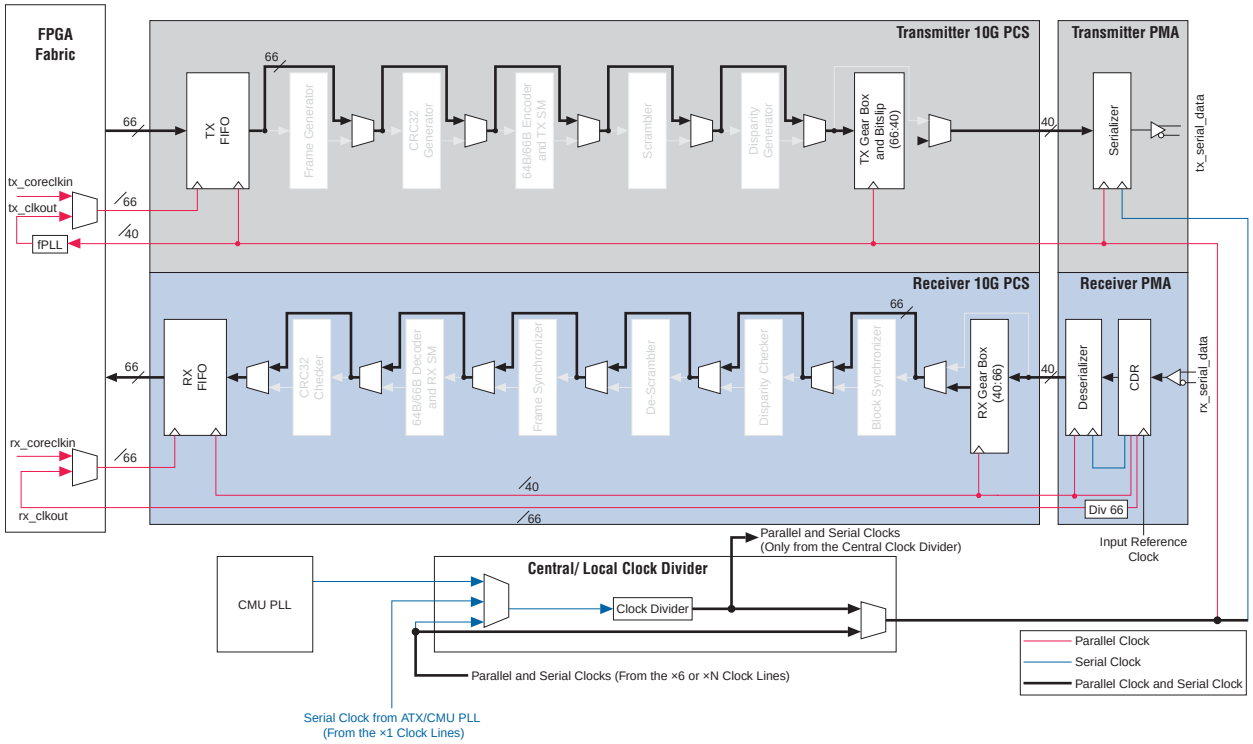
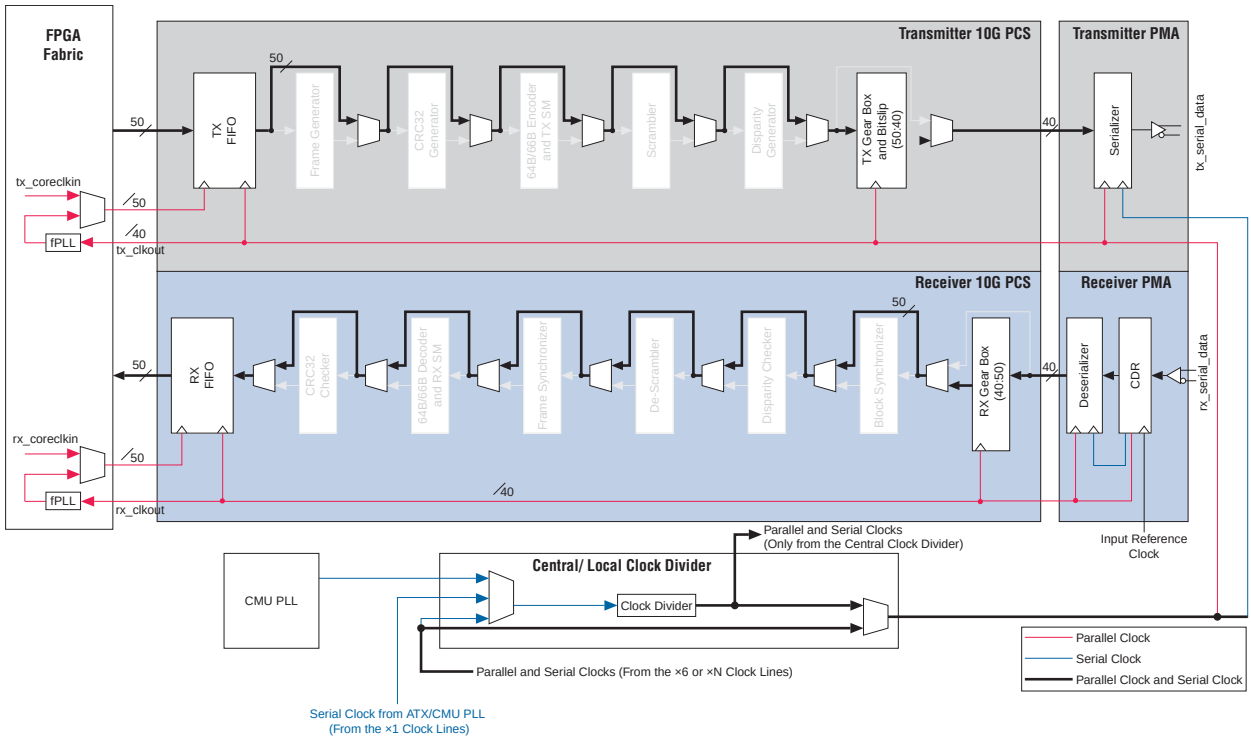


Figure 4–50 shows the clocking scheme when the gear box ratio (50:40) is not an integral multiple of the FPGA fabric interface width. For a gear box ratio of 50:40, you must instantiate a fractional PLL ("fPLL" in Figure 4–50) to provide the appropriate clock frequency to the write side of the TX FIFO. Set the division factor in the fractional PLL so that its output frequency is equal to the transmitter or lane data rate divided by 50 for the 50:40 gear box ratio. The clock source that provides the input reference clock to the fractional PLL and the CMU or ATX transmit PLL must be the same because the TX FIFO operates as a phase compensation FIFO, unlike a clock compensation or rate match FIFO. Therefore, the clock requires a zero ppm between the read and write operations.

For the receiver side, enable the `rx_coreclkin` port and connect a second fractional PLL output to the `rx_coreclkin` port. The RX FIFO operates as a phase compensation FIFO. Therefore, the read and write side of the RX FIFO must have a zero ppm difference.

Figure 4–50. 10G PCS Low Latency Datapath with the Gear Box Ratio of 50:40



Note to Figure 4–50:

(1) The clock source that provides the input reference clock to the fractional PLL (fPLL in Figure 4–50) and the CMU or ATX PLL (the CMU or ATX PLL generates the high-speed clock for the serializer) must be the same. The transmitter and the receiver FIFOs compensates only for phase differences. Therefore, the same clock source ensures zero ppm between the read and write clocks of the FIFOs.

Using the coreclkin Ports

The `tx_coreclkin` and `rx_coreclkin` ports offer the flexibility to use the `tx_clkout` and `rx_clkout` from one channel to clock the TX and RX FIFOs multiple channels for source synchronous links or if the upstream transmitters are all clocked by the same clock source. The `tx_coreclkin` and `rx_coreclkin` ports require a zero ppm difference between the `tx_clkout` and `rx_clkout` ports, respectively, with a divided-by-50 input frequency.

For more information about the core clocking scheme, refer to the following sections in the *Transceiver Clocking for Stratix V Devices* chapter:

- "User-Selected Transmitter Datapath Interface Clock"
- "User-Selected Receiver Datapath Interface Clock"

Merging Instances

You can merge transmitter and receiver instances with the different 10G PCS datapath configurations in the same 10 Gbps physical channel. For example, the Quartus II software allows you to create the two following instances and place them in the same physical transceiver channel:

- Transmitter only instance with a 40-bit FPGA fabric interface
- Receiver only instance with a 64-bit FPGA fabric interface

However, you cannot merge a transmitter instance and receiver instance (1 channel instance) using different PCS blocks (10G PCS and standard PCS) within the same physical transceiver channel.

Transceiver Channel Placement Guidelines

Stratix V devices allow the placement of up to four or five channels when a CMU PLL is used or up to six channels when an ATX PLL is used with Custom and Low Latency datapath configurations with Standard PCS and 10G PCS (same data rate) within the same transceiver bank.

Figure 4–42 on page 4–67 shows a non-bonded supported channel placement scenario using CMU and ATX PLLs. Figure 4–43 on page 4–68 shows a xN bonded supported channel placement scenario using CMU and ATX PLLs within a transceiver bank.

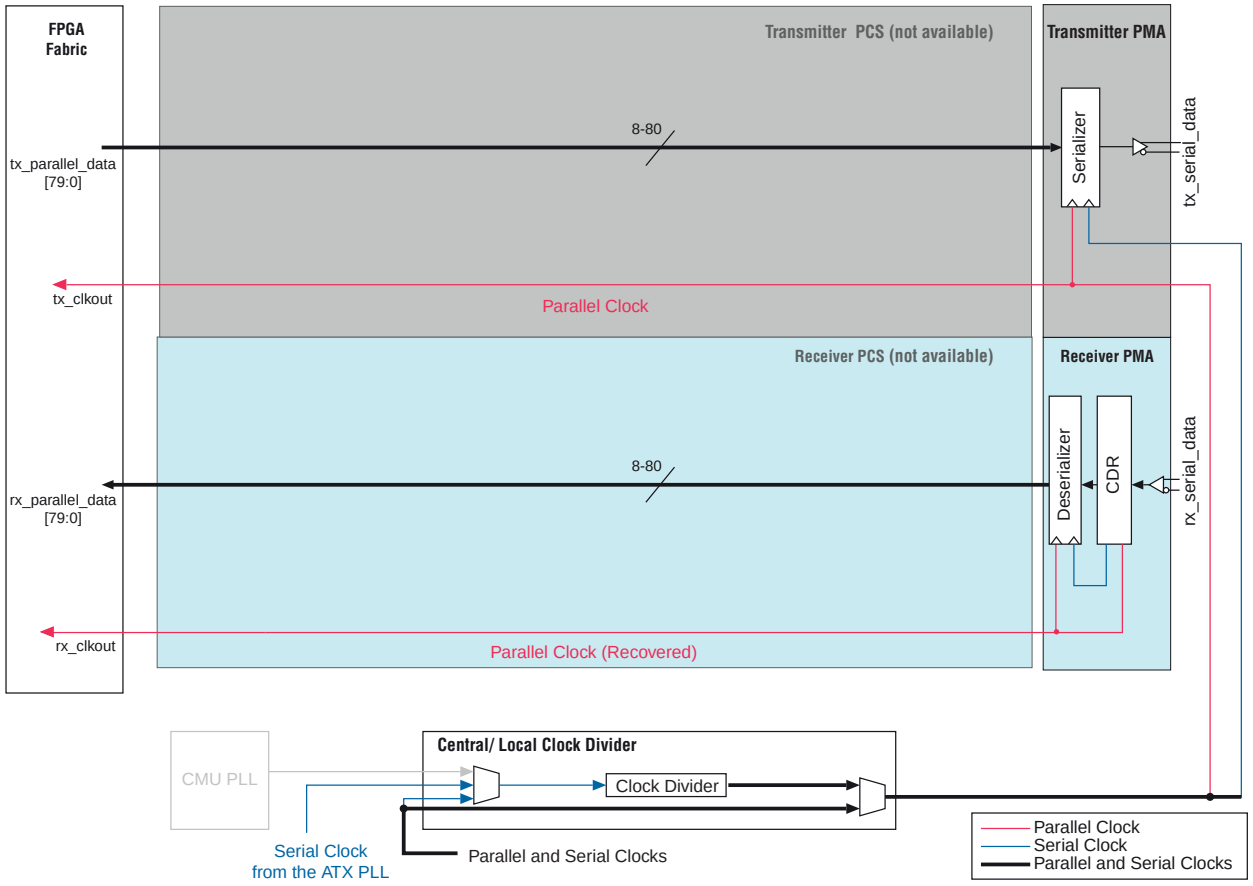
Stratix V GX Native PHY IP

The Stratix V GX transceivers can be used in the Custom datapath or Low Latency datapath configurations as described in "Stratix V Device Configurations" on page 4–54. The GX transceivers can also be configured in the PMA-Direct configuration using the Native PHY IP. In PMA-Direct configuration, there are no PCS blocks enabled in the datapath. Data and clock signals are interfaced directly to the transceiver PMA. You must also compensate for the timing and clock phase differences from the core fabric interface of the FPGA to the transceiver PMA. the PMA interface width has a wide range of selection from 8-bit, 10-bit, 16-bit, 20-bit, 32-bit, 40-bit, 64-bit, and 80-bit. The FPGA fabric interface width is fixed at 80-bit and you must select the correct ports for their PMA interface width configurations.

In the MegaWizard Plug-In Manager, you can implement a Native PHY link with the GX channel by instantiating the **Transceiver Native PHY IP** under **Transceiver PHY** in the Interfaces menu.

Figure 4–51 shows the transceiver datapath and clocking in Stratix V GX channels.

Figure 4–51. Transceiver Datapath in Stratix V GX Channels in PMA-Direct Configuration



For the clocking details of Stratix V GX channels, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

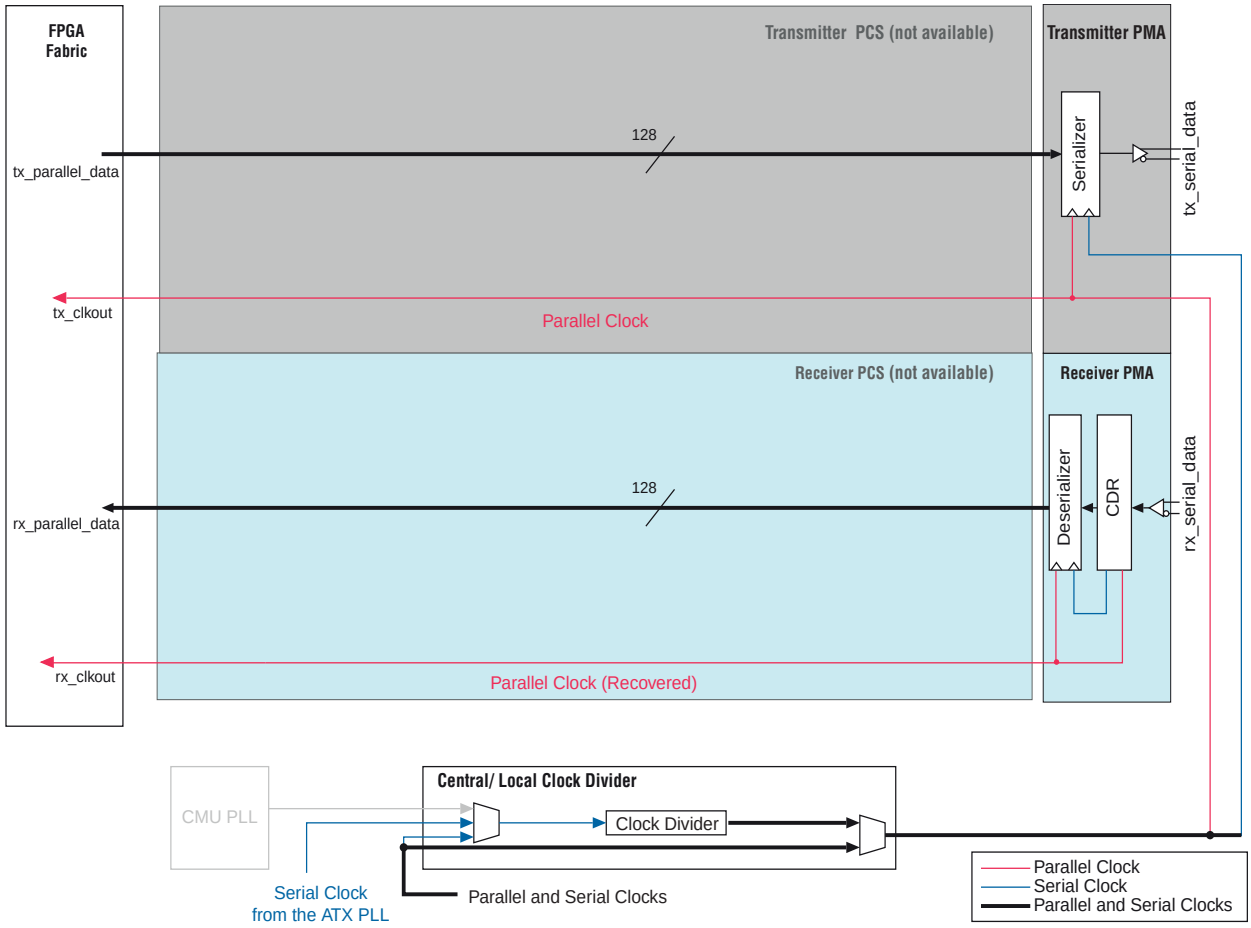
Stratix V GT Device Configurations

Stratix V GT devices contain both 28.05 Gbps GT transceivers and 12.5 Gbps GX transceivers. The GX transceivers can be configured as described in "Stratix V Device Configurations" on page 4–54. The GT transceivers can be configured only in PMA-Direct configuration using the Low Latency PHY IP. The GT channels can have serial data rates that range from 19.6 Gbps to 28.05 Gbps.

In the MegaWizard Plug-In Manager, you can implement a Low Latency PHY link with the GT channel by instantiating the **Low Latency PHY IP** under **Transceiver PHY** in the Interfaces menu. In the **Low Latency GUI** under the **General** tab, select **GT** on the **Datapath type** selection.

Figure 4–52 shows the transceiver datapath and clocking in Stratix V GT channels.

Figure 4–52. Transceiver Datapath in Stratix V GT Channels



For the clocking details of Stratix V GT channels, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

Document Revision History

Table 4–13 lists the revision history for this chapter.

Table 4–13. Document Revision History (Part 1 of 2)

Date	Version	Changes
June 2012	2.3	■ Added "CPRI and OBSAI—Deterministic Latency Protocols" section. ■ Added "Multi-Lane Deskew Alignment", "Transmit and Receive FIFO Control and Status", and "Transceiver Multi-Lane Bonding and Transmit Skew" sections to the Interlaken section. ■ Updated the "Transceiver Channel Placement Guidelines" section in the XAUI section. ■ Updated Figure 4–4, Figure 4–5, Figure 4–6, Figure 4–12, Figure 4–13, Figure 4–19, Figure 4–29, Figure 4–31, Figure 4–33, Figure 4–34, Figure 4–35, Figure 4–36, Figure 4–38, Figure 4–39, Figure 4–40, Figure 4–41, Figure 4–43, Figure 4–45. ■ Updated Table 4–1, Table 4–3, Table 4–4, Table 4–5, Table 4–10, Table 4–11, Table 4–12.
February 2012	2.2	■ Added Figure 4–38 and Figure 4–39. ■ Updated Figure 4–3, Figure 4–5, Figure 4–6, Figure 4–7, Figure 4–8, Figure 4–9, Figure 4–11, Figure 4–13, Figure 4–14, Figure 4–26, Figure 4–27, Figure 4–38, Figure 4–39, Figure 4–42, and Figure 4–45. ■ Added Table 4–4. ■ Updated Table 4–5. ■ Removed "Transceiver Channel Placement Guidelines" in Interlaken section. ■ Removed "Transceiver Channel Placement Guidelines" in 10GBASE-R section.
December 2011	2.1	■ Updated Figure 4–3, Figure 4–5, Figure 4–7, Figure 4–8, Figure 4–9, Figure 4–11, Figure 4–12, Figure 4–14, Figure 4–15, Figure 4–16, Figure 4–17, Figure 4–19, Figure 4–20, Figure 4–21, Figure 4–22, Figure 4–23, Figure 4–30, Figure 4–31, Figure 4–32, Figure 4–33, Figure 4–34, Figure 4–35, Figure 4–36, Figure 4–43, Figure 4–45, and Figure 4–48. ■ Updated Table 4–3 and Table 4–9. ■ Removed "Clock Compensation for Repeater Applications" section. ■ Minor text edits.
November 2011	2.0	■ Merged the 11.0 Transceiver Custom Configurations chapter into this chapter and reorganized this chapter. ■ Added PCI Gen3 information. ■ Added Stratix V GT device information. ■ Removed "GIGE" section.

Table 4–13. Document Revision History (Part 2 of 2)

Date	Version	Changes
May 2011	1.2	■ Updated Figure 4–8 and Figure 4–9. ■ Updated "Supported Features" on page 4–10. ■ Updated Table 4–5. ■ Updated Figure 4–16, Figure 4–18, and Figure 4–19. ■ Added "GIGE" section. ■ Updated "XAUI" on page 4–38. ■ Updated "Transceiver Datapath in a XAUI Configuration" on page 4–39. ■ Updated "Transceiver Channel Placement Guidelines" on page 4–44. ■ Updated Figure 4–33. ■ Chapter moved to volume 3 for the 11.0 release.
December 2010	1.1	■ Updated "PCI Express (PIPE) 2.0 Interface", "Dynamic Switching Between Gen1 (2.5 Gbps) and Gen2 (5 Gbps) Signal Rates", "Receiver Status", and "Receiver Detection" sections. ■ Updated Figure 4–32.
April 2010	1.0	Initial release.

5. Transceiver Loopback Support in Stratix V Devices

SV52007-2.4

This chapter describes the loopback options on Stratix® V devices, which allow you to verify how different functional blocks work in the transceiver (physical media attachment) PMA and the standard physical coding sublayer (PCS).

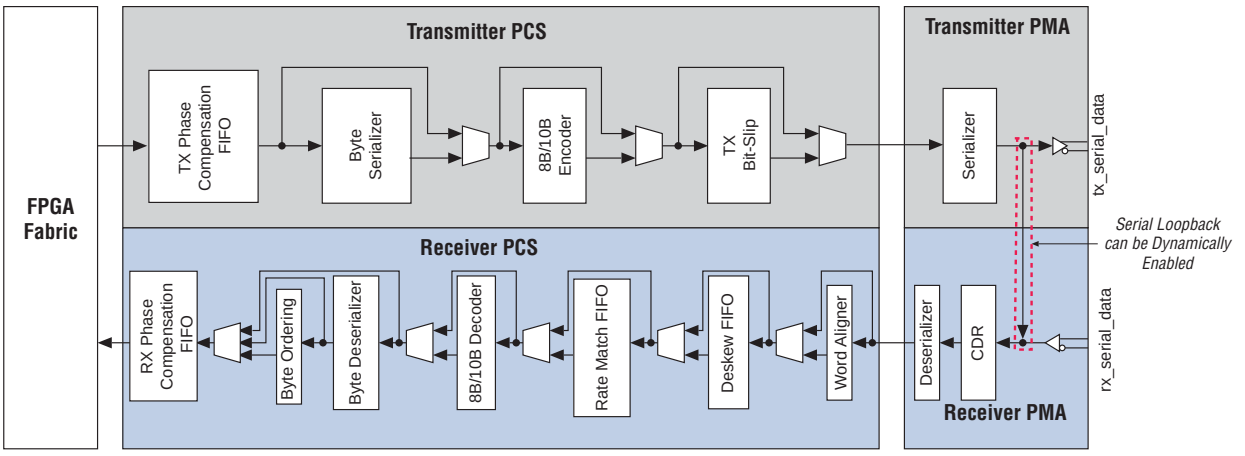
The four available loopback options are:

- "Serial Loopback"—available in all configurations except the PCI Express® (PCIe®) configuration
- "PCIe Reverse Parallel Loopback" on page 5–2—supported in the PCIe configuration only
- "Reverse Serial Loopback" on page 5–3—supported when you customize a configuration
- "Reverse Serial Pre-CDR Loopback" on page 5–4—supported when you customize a configuration

Serial Loopback

Figure 5–1 shows the datapath for serial loopback. The data from the FPGA fabric passes through the transmitter channel and is looped back to the receiver channel, bypassing the receiver buffer. The received data is available to the FPGA logic for verification. With this option, you can review how the enabled PCS and PMA functional blocks in the transmitter and receiver channel work. Furthermore, you can dynamically enable serial loopback on a channel-by-channel basis. You enable serial loopback by accessing the PHY IP core registers through the Avalon® Memory-Mapped PHY management interface.

Figure 5–1. Serial Loopback Datapath



© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

For further information about PHY IP core registers, refer to the *Altera Transceiver PHY IP Core User Guide*.

When you enable serial loopback, the transmitter channel sends the data to both the `tx_serial_data` output port and to the receiver channel. The differential output voltage on the `tx_serial_data` port is based on the selected differential output voltage (V_{OD}) settings. The looped back data is received by the receiver clock data recovery (CDR) and is retimed through different clock domains. You must provide an alignment pattern for the word aligner to enable the receiver channel to retrieve the byte boundary.

If the device is not in the serial loopback configuration and is receiving data from a remote device, the receiver CDRs recovered clock is locked to the data from that source. If the device is placed in the serial loopback configuration, the data source to the receiver changes from the remote device to the local transmitter channel, which prompts the receiver CDR to start tracking the phase of the new data source. During this time, the receiver CDRs recovered clock may be unstable. As the receiver PCS is running off of this recovered clock, you must reset the receiver PCS by asserting the `rx_digitalreset` signal during this time period.

When moving into or out of serial loopback, you must assert `rx_digitalreset` for a minimum of two parallel clock cycles.

For more information about asserting `rx_digitalreset`, refer to the *Altera Transceiver PHY IP Core User Guide*.

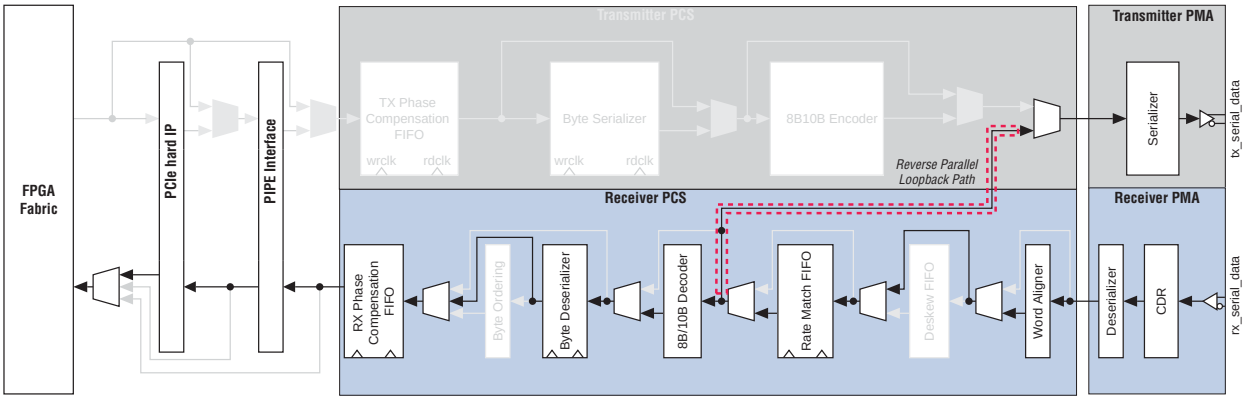
PCIe Reverse Parallel Loopback

PCIe reverse parallel loopback is available only in the PCIe configuration for Gen1 and Gen2 data rates. As shown in Figure 5–2, the received serial data passes through the receiver CDR, deserializer, word aligner, and rate matching FIFO buffer. The data is then looped back to the transmitter serializer and transmitted out through the `tx_serial_data` port. The received data is also available to the FPGA fabric through the `rx_parallel_data` signal. Reverse parallel loopback is designed for the PCIe specification 2.0. To enable this loopback configuration, assert the `tx_detectrxloopback` signal.

The PCIe reverse parallel loopback is the only loopback option supported in the PCIe configuration. PCIe reverse parallel loopback is not supported in the GT channels of Stratix V GT devices.

For more information, refer to the "PCI Express Reverse Parallel Loopback" section in the *Transceiver Configurations in Stratix V Devices* chapter.

Figure 5–2. PCIe Reverse Parallel Loopback Configuration Datapath *(1)*



Note to Figure 5–2:

(1) Grayed-out blocks are not active in this configuration.

Reverse Serial Loopback

You can enable reverse serial loopback by accessing the register space within the reconfiguration controller through the Avalon-MM interface.

For the register definitions needed to enable this functionality, refer to the *Altera Transceiver PHY IP Core User Guide*.

In reverse serial loopback, the data is received through the `rx_serial_data` port, retimed through the receiver CDR, and sent out to the `tx_serial_data` port. The received data is also available to the FPGA fabric through the `rx_parallel_data` signal. No dynamic pin control is available to select or deselect reverse serial loopback. Figure 5–3 shows the transceiver channel datapath for reverse serial loopback mode.

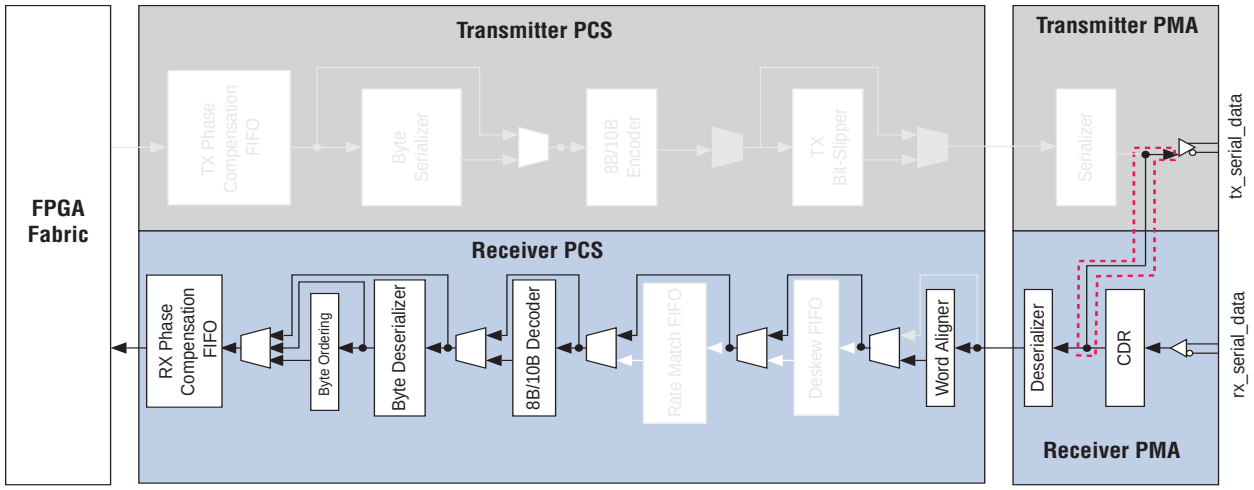
You set the reverse serial loopback with the PMA analog registers in the reconfiguration controller.

The only transmitter channel resource used when implementing reverse serial loopback is the transmitter buffer. You can define the V_{OD} and first post tap values on the transmitter buffer using assignment statements in the project **.qsf** or in the Quartus II Assignment Editor. You can also change these values dynamically with the reconfiguration controller.

For more information about how to dynamically change these analog settings, refer to the *Altera Transceiver PHY IP Core User Guide*.

Reverse serial loopback is often implemented when using an external bit error rate tester (BERT) on the upstream transmitter.

Figure 5–3. Reverse Serial Loopback Datapath *(1)*



Note to Figure 5–3:

(1) Grayed-out blocks are not active in this configuration.

Reverse Serial Pre-CDR Loopback

You can enable reverse serial pre-CDR loopback by accessing the register space within the reconfiguration controller through the Avalon-MM interface.

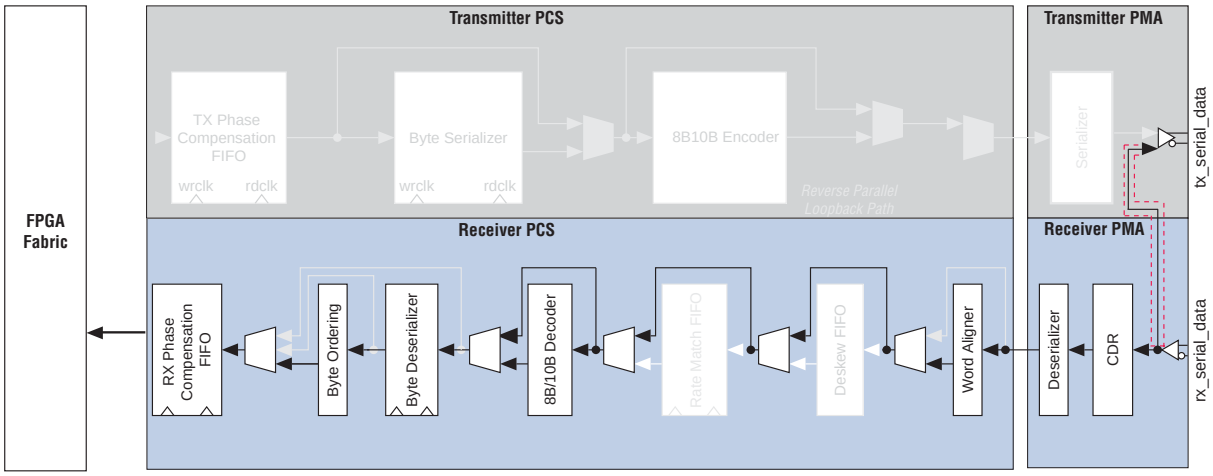
For the register definitions needed to enable this functionality, refer to the *Altera Transceiver PHY IP Core User Guide*.

In reverse serial pre-CDR loopback, the data received through the `rx_serial_data` port is looped back to the `tx_serial_data` port *before* the receiver CDR. The received data is also available to the FPGA fabric through the `rx_parallel_data` signal. No dynamic pin control is available to select or deselect reverse serial pre-CDR loopback. Figure 5–4 shows the transceiver channel datapath for reverse serial pre-CDR loopback mode.

You set the reverse serial pre-CDR loopback with the PMA analog registers in the reconfiguration controller.

The only transmitter channel resource used when implementing reverse serial pre-CDR loopback is the transmitter buffer. You can change the V_{OD} on the transmitter buffer through the Custom PHY IP MegaWizard™ Plug-In Manager. The receiver data characteristics that are looped back in reverse serial pre-CDR loopback are preserved by the transmitter buffer. The pre-emphasis settings for the transmitter buffer cannot be changed in this configuration.

Figure 5–4. Reverse Serial Pre-CDR Loopback Datapath *(1)*



Note to Figure 5–4:

(1) Grayed-out blocks are not active in this configuration.

Document Revision History

Table 5–1 lists the revision history for this chapter.

Table 5–1. Document Revision History

Date	Version	Changes
June 2012	2.4	Update for the Quartus® II software version 12.0
February 2012	2.3	Minor edits for clarity.
December 2011	2.2	Updated the document to clarify information.
November 2011	2.1	■ This chapter was formerly chapter 6. ■ There are no content changes for this version of the chapter.
May 2011	2.0	■ Added the “Reverse Serial Loopback” and “Reverse Serial Pre-CDR Loopback” sections. ■ Updated Figure 5–2. ■ Updated the chapter title. ■ Chapter moved to Volume 3. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

6. Dynamic Reconfiguration in Stratix V Devices

SV52008-2.3

This chapter describes the dynamic reconfiguration features available in Stratix® V transceivers.

For information about features that will be supported in a future release of the Quartus® II software, refer to the *Upcoming Stratix V Device Features* document.

Offset cancellation, as described in "Offset Cancellation Calibration" on page 6–2, is automatically performed during device power-up (except when Configuration via Protocol (CvP) is used). After the reset sequence is completed, the Transceiver Reconfiguration Controller is ready for user-controlled operations.

Stratix V transceivers have a dedicated reconfiguration space. You can enable or disable the following features in the Transceiver Reconfiguration Controller:

- "PMA Controls Reconfiguration" on page 6–2
- "On-Chip Signal Quality Monitoring (EyeQ)" on page 6–2
- "Decision Feedback Equalization" on page 6–3
- "Adaptive Equalization" on page 6–3

The transceiver supports three loopback modes, as described in "Enabling and Disabling Loopback Modes" on page 6–4:

- Serial loopback
- Reverse serial loopback
- Reverse serial pre-clock data recovery (CDR) loopback

The Transceiver Reconfiguration Controller supports two reconfiguration modes, as described in "Transceiver PLL Reconfiguration" on page 6–4:

- Channel
- Phase-locked loop (PLL)

For more information about the controller, its settings, and how to implement it in your design, refer to the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Offset Cancellation Calibration

Every transceiver channel in Stratix V devices has offset cancellation circuitry to compensate for non-optimal offsets caused by process variation.

The offset cancellation circuitry is controlled by the offset cancellation control logic IP within the Transceiver Reconfiguration Controller. This IP is active only during device power-up, and it automatically performs offset cancellation. After the reset sequence is completed, as indicated by the deassertion of the `reconfig_busy` signal, the Transceiver Reconfiguration Controller is ready for user-controlled operations.

The clock (`mgmt_clk_clk`) to the Transceiver Reconfiguration Controller is also used for transceiver calibration and should be within the range of 100-125 MHz. If the clock (`mgmt_clk_clk`) is not free-running, the reconfiguration controller reset (`mgmt_rst_reset`) must be held in reset until the clock is stable.

PMA Controls Reconfiguration

Use the Transceiver Reconfiguration Controller to reconfigure the following analog settings:

- Differential output voltage (V_{OD})
- Pre-emphasis taps
- Receiver equalization control (AC gain)
- Receiver equalization DC gain

The dynamic reconfiguration controller in Stratix V devices provides an Avalon® Memory-Mapped (Avalon-MM) user interface to dynamically reconfigure individual PMA settings on a per channel basis.

For more information about reconfiguring a specific PMA control, refer to "PMA Analog Controls" in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

For a specific design example, refer to *AN 645: Dynamic Reconfiguration of PMA Controls in Stratix V Devices*.

On-Chip Signal Quality Monitoring (EyeQ)

The bit error rate (BER) eye contour can be used to measure the quality of the received data. EyeQ is a debug and diagnostic tool that analyzes the received data recovery path, including the receiver's gain, noise level, and recovery clock jitter. EyeQ can also measure vertical eye height, effectively allowing a BER eye contour to be plotted.

EyeQ uses a phase interpolator (PI) and sampler (SMP) to estimate the horizontal eye opening. Controlled by a logic generator, the PI generates a sampling clock and the SMP samples the data from the receiver output. The SMP outputs parallel data that is monitored for CRC or BER errors. When the PI output clock phase is shifted by small increments, the data error rate goes from high to low to high if the receiver is good. The number of steps of valid data is defined as the width of the eye. If none of the steps yield valid data, the width of the eye is equal to 0, which means the eye is closed.

The Transceiver Reconfiguration Controller in Stratix V devices provides an Avalon-MM user interface to enable the EyeQ feature and step through the 32 phase steps of the recovered clock. You can use EyeQ to quantify the change in eye opening for different analog settings.

For more information about enabling the EyeQ feature, refer to the "EyeQ" section in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

Decision Feedback Equalization

Decision feedback equalization (DFE) helps compensate for backplane attenuation because of insufficient bandwidth. DFE works by estimating the intersymbol interference (ISI) that is imposed by the channel on an incoming bit and canceling out the ISI as that bit is sampled by the CDR circuitry. The advantage of DFE is that it boosts the power of the highest frequency component of the received data without increasing its noise power. Use DFE in conjunction with the transmitter pre-emphasis and receiver linear equalization.

The Transceiver Reconfiguration Controller in Stratix V devices provides an Avalon-MM user interface to step through the DFE tap settings.

In Stratix V GT devices, the GT channels do not support DFE. This feature is supported in the GX channels.

For more information about DFE, refer to the "DFE" section in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

Adaptive Equalization

High-speed interface systems require different equalization settings to compensate for changing data rates and backplane losses. Manual tuning of the receiver channel equalization stages involves finding the optimal settings through trial and error, and then locking in those values during compilation. This manual static method is cumbersome and inefficient when system characteristics vary. The adaptive equalization (AEQ) feature solves this problem by automatically tuning an active receiver channel equalization filter based on a frequency content comparison between the incoming signals and the internally generated reference signals.

In Stratix V GT devices, the GT channels do not support AEQ. This feature is supported in the GX channels.

The Transceiver Reconfiguration Controller in Stratix V devices provides an Avalon-MM user interface to enable the AEQ feature. Refer to the "AEQ" section in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide* for more information about enabling different options and using them to control the AEQ hardware.

Enabling and Disabling Loopback Modes

You can enable or disable various loopback modes by writing the appropriate PHY management interface or Transceiver Reconfiguration Controller registers. You can use the different loopback modes to help debug your design.

The following loopback paths are available:

- **Serial loopback path**—This mode takes the output at the serializer and feeds it back to the CDR. While in this mode, the serializer also feeds the data to the TX output port. Enabling or disabling serial loopback mode is done through the PHY management interface.
- **Reverse serial loopback path**—The receiver (RX) captures the input data and feeds it into the CDR. The recovered data from the CDR output feeds into the TX driver and sends the data to the TX pins through the TX driver. For this path, you can test the RX and CDR. In the reverse serial loopback mode, you can program the TX driver to use either the main tap only or the main tap and the pre-emphasis first post-tap. Enabling or disabling reverse serial loopback mode is done through the PMA analog control registers in the Transceiver Reconfiguration Controller.
- **Reverse serial pre-CDR loopback path**—The RX captures the input data and feeds it back to the TX driver through a buffer. This path does not go through the CDR. With this path, you can perform a quick check for the quality of the RX and TX buffers. Enabling or disabling the reverse serial pre-CDR loopback mode is done through the PMA analog control registers in the Transceiver Reconfiguration Controller.

For descriptions of the different loopback modes, refer to the *Transceiver Loopback Support in Stratix V Devices* chapter. For implementation details about the different loopback modes, refer to "Loopback Modes" in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

Transceiver PLL Reconfiguration

The Stratix V Transceiver Reconfiguration Controller supports PLL reconfiguration. You can reconfigure the PLLs in the following ways:

- Switch the reference clocks to the CDR and transmitter PLLs
- Reconfigure the PLL connected to the transceiver channel

For example, you can switch the reference clock from 100 MHz to 125 MHz. You can also change the data rate from 2.5G to 5G by reconfiguring the transmitter PLL connected to the transceiver channel.

The Transceiver Reconfiguration Controller in Stratix V devices provides an Avalon-MM user interface to perform PLL reconfiguration.

For more information about performing PLL reconfiguration, refer to the "PLL Reconfiguration" section in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

Transceiver Channel and Interface Reconfiguration

The Stratix V Transceiver Reconfiguration Controller supports channel and interface reconfiguration.

Transceiver Channel Reconfiguration

You can reconfigure the channels in the following ways:

- Reconfigure the CDR of the receiver channel.
- Enable and disable the static PCS blocks.
- Select an alternate PLL in the transceiver block to supply a different clock to the transceiver clock generation block.
- Every transmitter channel has a clock divider. When you reconfigure these clock dividers, ensure that the functional mode of the transceiver channel supports the reconfigured data rate.

For example, you can change the data rate from 6.25G to 3.125G by reconfiguring the clock divider.

Transceiver Interface Reconfiguration

You can reconfigure the transceiver interfaces by reconfiguring the FPGA fabric-transceiver channel data width. This feature is available for both the Custom and Low Latency PHY IP cores.

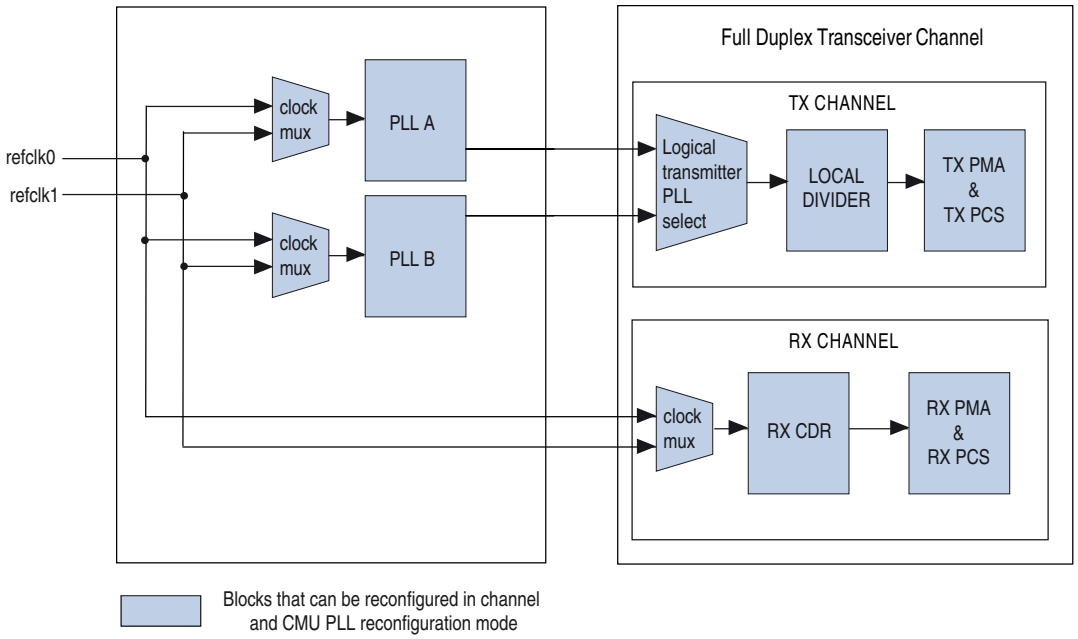
Channel reconfiguration only affects the channel involved in the reconfiguration (the transceiver channel specified by the unique logical channel address), without affecting the remaining transceiver channels controlled by the same Transceiver Reconfiguration Controller. PLL reconfiguration affects all channels that are currently using that PLL for transmission.

Channel reconfiguration from either a transmitter-only configuration to a receiver-only configuration or vice versa is not allowed.

The blocks that are reconfigured by this dynamic reconfiguration mode are the PCS and PMA blocks of a transceiver channel, the clock divider settings of the transmitter and receiver channel, and the CDR and transmitter PLL.

Figure 6–1 shows the functional blocks that you can dynamically reconfigure using transceiver channel and PLL reconfiguration mode.

Figure 6–1. Transceiver Channel and PLL Reconfiguration in a Transceiver Block



For more information about the transceiver channel and PLL reconfiguration, refer to the "Channel and PLL Reconfiguration" section in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

Document Revision History

Table 6–1 lists the revision history for this chapter.

Table 6–1. Document Revision History (Part 1 of 2)

Date	Version	Changes
June 2012	2.3	Minor text edits.
February 2012	2.2	■ Updated the "Decision Feedback Equalization" section. ■ Minor text edits.
December 2011	2.1	Minor text edits.
November 2011	2.0	■ Added "Offset Cancellation Calibration", "On-Chip Signal Quality Monitoring (EyeQ)", "Decision Feedback Equalization", "Adaptive Equalization", "Enabling and Disabling Loopback Modes", "Transceiver PLL Reconfiguration", and "Transceiver Channel and Interface Reconfiguration" sections. ■ Minor text edits.
May 2011	1.2	■ Deleted "Indirect Addressing" section and part of introduction. ■ Added link to AN 645. ■ Chapter moved to Volume 3.

Table 6–1. Document Revision History (Part 2 of 2)

Date	Version	Changes
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

This chapter provides additional information about the document and Altera.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Nontechnical support (general)	Email	nacomp@altera.com
(software licensing)	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *Stratix IV Design Guidelines*.
italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, *<file name>* and *<project name>***.pof** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, "Typographic Conventions."

Visual Cue	Meaning
`Courier type`	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. The suffix n denotes an active-low signal. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
↵	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	The question mark directs you to a software help system with related information.
	The feet direct you to another document or website with related information.
	The multimedia icon directs you to a related multimedia presentation.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.